UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Robust human-robot collaboration for polishing tasks in the automotive industry

PhD. Thesis

Author: **Alberto García Fernández**

Supervisors: **Luis Ignacio Gracia Calandín**
Josep Tornero Montserrat

PhD. in Automation, Robotics and Industrial Computer Science

Universitat Politècnica de València

Valencia, July 2023

This work has been financed by the Generalitat Valenciana (ACIF 19 program) and the Universitat Politècnica de València (FPI program, subprogram 1), via their corresponding *Formación de Personal Investigador (FPI)* grants for predoctoral researchers.

A mi padre, que vio empezar esta tesis y no llegó a verla terminar.

Agradecimientos

Me gustaría empezar por agradecerle esto a mi familia.

Mi padre, Amador, una persona maravillosa y una mente brillante que ya no está entre nosotros, y mi madre, Carmina, una persona entregada y cariñosa que ha sido capaz de conseguir los mayores éxitos académicos a la vez que nos ha cuidado y cuida todavía; ellos me han animado a no dejar nunca de aprender y a no tirar la toalla. Mi hermana, Teresa, que ha heredado todo lo bueno de nuestros padres y, de cara a la tesis, me ha ayudado a relativizar y ordenar mis prioridades, sin lo cual no hubiera podido concluirla.

A mis abuelos: Concha, que sigue siendo joven en ánimo y salud pese a las adversidades; Amador y Herminio, que trabajaron toda su vida para que hoy mi familia pueda vivir con tranquilidad; y Enriqueta, que no pudo llegar a conocerme y, aun así, supo que todos saldríamos adelante.

A mis tíos y tías, a mis primas; y no puedo olvidarme de Amparo. De no ser por ella, me hubiera olvidado la cabeza cada mañana al ir a clase.

Por supuesto, como parte de una familia más amplia, a mi pareja, Anna, y a mis amigos. Tengo muchos amigos, algo que me llena de alegría, pero hay algunos que me han acompañado desde el inicio de mi vida en la universidad y quiero nombrarlos: Carlos, Álex Fuster, Claudia, Blai, Bryan, Ana y Ernest.

En segundo lugar, quiero agradecerles a mis directores Josep y Luis por su guía y los conocimientos que me han transmitido, a Ernesto, que ha sido mi ángel guardián en el día a día del trabajo de tesis, a Vicent Girbés por compartir conmigo su experiencia y ahorrarme sufrimientos, a Mario y Vicent Franch por toda su ayuda y paciencia, y a todo el personal del IDF, que se ha volcado conmigo.

Y, por último, a mis compañeros de TCI, que me han dado fuerzas para terminar la redacción de la tesis a la vez que emprendía mi carrera laboral fuera de la universidad.

Resum

La present tesi aborda la Interacció Humà-Robot per a tasques industrials de tractament de superfícies, amb l'objectiu d'obtindre una veritable sinergia entre l'operador humà i el sistema robotitzat, així com un funcionament robust. En concret, la tesi estableix les bases sobre les quals un robot amb assistència per teleoperació o autònomament interacciona amb els humans i amb els altres robots a la zona de treball. Les propostes es validen mitjançant experimentació real utilitzant fins a dos robots manipuladors 6R i 7R, respectivament.

Les principals contribucions són:

– **Assistència robòtica per al poliment industrial amb aproximació suau a la superfície i restriccions de límit:** a l'aplicació desenvolupada en aquest capítol l'operador humà proporciona flexibilitat, guiant l'eina del sistema robotitzat per tal de tractar regions arbitràries de la superfície de la peça de treball; mentre que el sistema robotitzat proporciona força, precisió i seguretat, no només sostenint l'eina i mantenint la correcta orientació de la mateixa, sinó també garantint una aproximació suau a la superfície i confinant l'eina dins l'àrea permesa propera a la peça de treball. A més, quan l'usuari no està guiant l'eina de treball, el mode automàtic s'activa, de manera que el robot tracta àrees pre-establertes de la superfície.

– **Control de robots bimanuals emprant teleoperació assistida per a tasques de tractament superficial:** aquest capítol presenta una arquitectura de control per a un sistea robòtic bimanual, és a dir, dos braços robot desenvolupant una tasca cooperativament, amb l'objectiu de realitzar un tractament superficial amb el qual l'usuari humà teleopera parcialment ambdós braços. En particular, un braç robot, anomenat Robot de la Peça de Treball (RPT), sosté la peça de treball, mentre que

l'altre braç robot, anomenat Robot de Tractament Superficial (RTS), té l'eina de tractament superficial adjuntada al seu efector final. D'aquesta manera, algunes coordenades dels robots són teleoperades per l'usuari humà, mentre que les coordenades restants són controlades automàticament. En particular, l'usuari teleopera les sis coordenades del RPT per tal de posicionar la peça de treball en una posició i orientació adequades per a la tasca. A més, el teleoperador comanda dues coordenades linials del RTS per moure l'eina sobre la superfície de la peça de treball amb l'objectiu de realitzar un tractament superficial. Per tal d'assistir l'usuari humà durant la teleoperació, es defineixen una sèrie de restriccions per a ambdós braços robot, amb l'objectiu d'evitar excedir l'espai de treball permès. Adicionalment, un sensor de Força/Parell (F/P) fixat a l'efector final del RTS s'empra per adaptar automàticament l'eina del RTS de manera que s'abastisca la pressió desitjada entre l'eina i la peça de treball, així com per tal de mantindre l'orientació ortogonal de l'eina sobre la superfície de la peça. Aquesta aproximació és validada mitjançant experimentació real amb un braç robot industrial 6R i un cobot 7R.

– **Intefaç basada en Realitat Augmentada per a la teleoperació de robots bimanuals:** Aquest capítol presenta una interfaç original basada en realitat augmentada per teleoperar robots bimanuals, amb l'objectiu de superar els problemes de les interfaces convencionals basades en l'ús de PC. La interfaç proposada és més natural per a l'usuari, permetent-li veure la informació rellevant en forma d'hologrames alhora que veu en tot moment els elements reals involucrats en la tasca: robots, peces de treball, eines, etc. Adicionalment, aquest treball proposa i segueix una nova metodologia per dissenyar i desenvolupar interfaces de RA per a sistemes robòtics bimanuals. L'efectivitat i aplicabilitat de la interfaç de RA proposada es mostren mitjançant l'experimentació real amb una aplicació de robòtica bimanual avançada que consta de dos braços robot: un cobot 7R i un manipulador industrial 6R.

L'aplicació de tècniques de Control en Mode Lliscant (SMC, per les seues sigles en anglés)tant convencional com no convencional i una arquitectura de control basada en prioritats són les eines clau pel desenvolupament d'aquestes contribucions.

Resumen

La presente tesis aborda la Interacción Humano-Robot para tareas industriales de tratamiento superficial, con el objetivo de obtener una verdadera sinergia entre el operador humano y el sistema robotizado, así como un funcionamiento robusto. En concreto, la tesis establece las bases sobre las cuales un robot con asistencia por teleoperación o autónomamente interacciona con los humanos y con los otros robots en la zona de trabajo. Las propuestas se validan mediante experimentación real utilizando hasta dos robots manipuladores 6R y 7R respectivamente.

Las principales contribuciones son:

– **Asistencia robótica para el lijado industrial con aproximación suave a la superficie y restricciones de límite:** en la aplicación desarrollada en este capítulo el operador humano proporciona flexibilidad, guiando la herramienta del sistema robotizado para tratar regiones arbitrarias de la superficie de la pieza de trabajo; mientras que el sistema robotizado proporciona fuerza, precisión y seguridad, no solo sosteniendo la herramienta y manteniendo la correcta orientación de la misma, sino también garantizando una aproximación suave a la superficie y confinando la herramienta dentro del área permitida cercana a la pieza de trabajo. Además, cuando el usuario no está guiando la herramienta de trabajo, el modo automático se activa, de manera que el robot trata áreas pre-establecidas de la superficie.

– **Control de robots bimanuales usando teleoperación asistida para tareas de tratamiento superficial:** este capítulo presenta una arquitectura de control para un sistema robótico bimanual, esto es, dos brazos robot desempeñando una tarea cooperativamente, con el objetivo de realizar un tratamiento superficial en el cual el usuario humano tele-

opera parcialmente ambos brazos robot. En particular, un brazo robot, llamado Robot de la Pieza de Trabajo (RPT), sostiene la pieza de trabajo, mientras que el otro brazo robot, llamado Robot de Tratamiento Superficial (RTS), tiene la herramienta de tratamiento superficia sujeta a su efector final. De esta manera, algunas coordenadas de los robots son teleoperadas por el usuario humano, mientras que las coordenadas restantes son controladas automáticamente. En particular, el usuario teleopera las seis coordenadas del RPT para poner la pieza de trabajo en una posición y orientación adecuadas para la tarea. Además, el teleoperador comanda dos coordenadas lineales del RTS para mover la herramienta sobre la superficie de la pieza de trabajo con el objetivo de realizar el tratamiento superficial. Para asistir al usuario humano durante la teleoperación, se definen una serie de restricciones para ambos brazos robot, con el objetivo de evitar exceder el espacio de trabajo permitido. Adicionalmente, un sensor de Fuerza/Par (F/P) fijado al efector final del RTS se usa para adaptar automáticamente la herramienta del RTS de modo que se alcance la presión deseada entre la herramienta y la pieza de trabajo, así como para mantener la orientación ortogonal de la herramienta sobre la superficie de la pieza. Esta aproximación es validada mediante experimentación real con un brazo robot industrial 6R y un cobot 7R.

– **Interfaz basada en Realidad Aumentada para la teleoperación de robots bimanuales:** Este capítulo presenta una interfaz original basada en realidad aumentada para teleoperar robots bimanuales, con el objetivo de superar los problemas de las interfaces convencionales basadas en el uso de PC. La interfaz propuesta es más natural para el usuario, permitiéndole ver la información relevante en forma de hologramas a la vez que ve en todo momento los elementos reales involucrados en la tarea: robots, pieza de trabajo, herramienta, etc. Adicionalmente, este trabajo propone y sigue una nueva metodología para diseñar y desarrollar interfaces de RA para sistemas robóticos bimanuales. La efectividad y aplicabilidad de la interfaz de RA propuesta se muestran mediante la experimentación real con una aplicación de robótica bimanual avanzada que consta de dos brazos robot: un cobot 7R y un manipulador industrial 6R.

La aplicación de técnicas de Control en Modo Deslizante (SMC, por sus

siglas en inglés) tanto convencional como no convencional y una arquitectura de control basada en prioridades son las herramientas clave para el desarrollo de estas contribuciones.

Abstract

The present thesis work addresses Human-Robot Interaction for industrial surface treatment tasks, aiming to attain a true synergy between the human operator and the robot system, as well as a robust performance. Specifically, this thesis establishes the basis on which a robot, either assisted by teleoperation or working autonomously, interacts with humans and with other robots in its working area. These proposals are validated through real experimentation using up to two robot manipulators, 6R and 7R respectively.

The main contributions are:

– **Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints:** in the application developed in this chapter the human operator provides flexibility, guiding the tool of the robot system to treat arbitrary regions of the workpiece surface; while the robot system provides strength, accuracy and security, not only holding the tool and keeping the right tool orientation, but also guaranteeing a smooth approach to the workpiece and confining the tool within the allowed area close to the workpiece. Furthermore, when the user is not guiding the robot tool, the automatic mode is activated, so the robot treats pre-stablished areas of the surface. Moreover, a camera network is used to get a global view of the robot workspace in order to obtain the workpiece location accurately and in real-time. The effectiveness of the proposed approach is shown with several experiments using a 6R robotic arm.

– **Bimanual robot control using assisted teleoperation for surface treatment tasks:** this chapter presents a control architecture for a bimanual robotic system, i.e., two robot arms cooperatively performing the task, in order to conduct a surface treatment task in which the

human user partially teleoperates both robot arms. In particular, one robot arm, namely Workpiece Robot (WR), holds the workpiece while the other robot arm, namely Surface Treatment Robot (STR), has the treatment tool attached to its end-effector. In this way, some robot coordinates are teleoperated by the human user, while the remaining robot coordinates are automatically controlled. In particular, the user teleoperates all the six coordinates of the WR in order to put the workpiece in a proper position and orientation for the task. Moreover, the teleoperator commands two linear coordinates of the STR to move the treatment tool on the workpiece surface in order to apply the surface treatment. In order to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace. Furthermore, a Force/Torque (F/T) sensor attached to the STR end-effector is used to automatically adapt the STR tool in order to attain the desired pressure between the tool and the workpiece as well as to keep the tool orientation orthogonal to the workpiece surface. This approach is validated through real experimentation with an industrial 6R robotic arm and a 7R cobot.

– **Augmented reality-based interface for bimanual robot teleoperation:**This chapter presents an original augmented reality-based interface for teleoperating bimanual robots, in order to overcome the problems of the conventional PC-based interface. The proposed interface is more natural to the user, allowing him or her to see the relevant information in the form of holograms while still seeing at all times the real elements involved in the task: robots, workpiece, tool, etc. In addition, this work proposes and follows a new methodology to design and develop AR interfaces for bimanual robotic systems.The effectiveness and applicability of the proposed AR interface are shown by means of real experimentation with an advanced bimanual robot application consisting of two robotic arms: a 7R cobot and a 6R industrial manipulator.

The appliance of conventional and non-conventional Sliding Mode Control (SMC) techniques and a priority-based control architecture are the fundamental tools for the development of these contributions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automation is a growing tendency in all aspects of society, since it increases the speed, precision and efficiency with which tasks are solved, while also relieving human operators from hard, dangerous or repetitive aspects of said tasks.

Nonetheless, many operations cannot be fully automated yet or even rely mainly on manual labor and the personal skills of the operator, as in the case of surface treatment tasks in quality control of surfaces [Martínez and Ortega (2013)], as they require the abilities of a human operator. In these cases, Human-Robot Interaction (HRI) appears as a step forward, trying to combine the accuracy and strength of a robotic system and the adaptability of a human operator.

This interaction can be materialized as the synergistic cooperation between a robotic system and a human operator sharing the same workspace, but also as the teleoperation of the robotic system by a remote human user, in order to eliminate unnecessary or unfeasible human presence or direct intervention, thus increasing safety and also efficiency in regards to space requirements.

Moreover, the fact that the robotic system ensures the accomplishment of certain task requirements, allows the human not just to improve his or her precission or strength while carrying out formerly non-automated tasks, but also to command more complex robotic systems, such as bimanual systems, and carry out tasks which would not be possible if carried out manually, but can benefit from the intervention of a human operator.

Motivation

Human-Robot Interaction must deal with several challenging requirements for the robotic system and the control algorithm:

- Adapting to the unpredictability of the human operator's behaviour.
- Being intuitive enough so the human operator can understand it and interact with it.
- Integrating and processing information coming from many sources (i.e. the human operator, other robots and peripherals), in real-time.
- Enforcing task constraints in a way which guides and limits but does not erase the human initiative.
- Optimizing control tasks, so that different control laws are ensured in a hierarchical way.

However, once this requirements have been met, shared control between the robotic system and the human operator allows the user to overcome challenging tasks which require both dexterity and strength if performed manually. The application chosen in this thesis is surface treatment, such as sanding or deburring, which is currently a part of the quality control process in the automotive industry, among others, which remains mainly non-automated.

This application imposes constraints both when the tool approaches the surface, phase of the process in which a smooth movement towards the surface is needed, and when the tool is already in contact with the surface, phase in which keeping the pressure constant and the tool orthogonal to the surface becomes the main issue. As it is developed in the following chapters, several other functionalities can be added to this application in order reduce the user's required dexterity, such as artificial vision systems in order to automatically locate defects and completely automate part of the process or boundaries to make the application safer.

Furthermore, the same principles applied to Human-Robot Interaction can be generalized and applied to interaction between two robotic systems, developing a bimanual robotic system with which the user can, in turn, interact, with a shared control architecture, adding a new layer to the issue.

Finally, as the robotic system becomes more complex, so does the interaction with it, so finding how to make it easier to control for the human user is an opportunity to explore the possibilities of Augmented Reality.

All these issues motivate this research, which has achieved its goal with this present thesis document and its related contributions.

This research has led to four publications in journals of high scientific impact, with the author of this PhD thesis as first author. These publications will be referenced more completely further on, but can be shortly listed as follows:

- "Human-Robot Cooperation for Surface Repair Combining Automatic and Manual Modes," in IEEE Access
- "Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints", Computers & Industrial Engineering
- "Bimanual robot control for surface treatment tasks", International Journal of Systems Science
- "Augmented Reality-Based Interface for Bimanual Robot Teleoperation", Applied Sciences

Thesis Outline

The present document is structured in seven chapters, including this one and the conclusion chapter:

- **Chapter 2: State of the art.** This chapter reviews the fundamentals about Human-Robot Interaction, assisted teleoperation, augmented reality, the surface treatment task, bimanual robotics, task optimization and Sliding Mode Control. Note that this is a generic addressing of the state of the art, which is concretized in Chapters 4, 5 and 6.
- **Chapter 3: Theoretical basis.** In this chapter, the mathematical basis of robot kinematics, task-priority based control strategies, Sliding Mode Control (both conventional and non-conventional) and the geometrical boundary models are discussed.
- **Chapter 4: Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints.** This chapter presents the first application developed in this thesis. In particular, in the application the human operator guides the tool of the robot system to treat arbitrary regions of the workpiece surface; while the robot system holds the tool keeps the right tool orientation and also

guarantees a smooth approach to the workpiece and confines the tool within the allowed area close to the workpiece. Moreover, when the user is not guiding the robot tool, a robot automatic operation is activated to perform the treatment in prior established regions. Furthermore, a camera network is used to get a global view of the robot workspace in order to obtain the workpiece location accurately and in real-time.

- **Chapter 5: Bimanual robot control using assisted teleoperation for surface treatment tasks.** In this chapter, the second main application of this thesis is discussed. It develops a method to perform surface treatment tasks using a bimanual robotic system, where one robot arm holds the workpiece and the other holds the surface treatment tool. Some robot coordinates from each robot arm are teleoperated by the human user, while the remaining ones are automatically controlled. Furthermore, the surface treatment robot reacts to the workpiece robot movements using the information obtained from a Force-Torque sensor, keeping contact and orthogonality with the workpice surface. In addition, several other constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace.

- **Chapter 6: Augmented reality-based interface for bimanual robot teleoperation.** This chapter presents a novel methodology for the design of interfaces based on augmented reality so that the interaction of human users with bimanual robot systems is natural and intuitive. In this way, an Augmented-Reality Interface to control bimanual robot systems is developed, taking as its case of application the complex application presented in Chapter 5.

- **Chapter 7: Conclusions**. This chapter presents the conclusions of the present research and references its four published contributions, reviewing their scientific impact.

- **Bibliography**.

In the next chapter, a state of the art of the fields of research involved in the thesis is presented.

Chapter 2

State of the Art

In this chapter, previous research, concepts and tools used in this work are presented. Furthermore, each chapter will analyze in detail its own state of the art.

2.1 Human Robot Interaction

As presented in 1, there are tasks that cannot be fully automated yet, as is the case of study in this work, surface treatment, so Human-Robot Interaction offers the possibility of combining the knowledge and adaptability of the human-operator with the strength and accuracy of a robotic system.

Moreover, even in cases of automated tasks with sophisticated AI, autonomous systems can be improved by working together with a human operator, that is, introducing shared-control architectures (Johnson and Vera, 2019).

Human-Robot Interaction can be developed as a cooperation between a robotic system and a human user sharing the same workspace or, on the other hand, as a remote interaction, an assisted teleoperation. A review of both applications of Human-Robot Interactions in the scope of this work is presented below.

2.1.1 Human Robot Cooperation

Most of the previous research that considers a robot system to perform surface treatment tasks uses an *automatic mode* of operation, i.e., no kind of

cooperation or interaction was considered between the human operator and the robot system to perform the surface treatment task. This is the case of works (Villagrossi et al., 2017; Diao et al., 2018; Mohsin et al., 2017; Mohsin et al., 2019; Oba and Kakinuma, 2017; Tian et al., 2016; Sato et al., 2017; Segreto and Teti, 2019; Huang et al., 2020; Kharidege et al., 2017).

In contrast to the works mentioned above, the proposed approach develops, from a qualitative perspective, a synergistic cooperation between the human operator and the robot system. However, the degree of interaction in the field of human-robot cooperation is not homogeneous, and the majority of previous research in this field tends to present a rather limited degree of interaction.

Some of the human-robot cooperation contributions limit the interaction to an offline task previous to the actual online surface treatment operation, as is the case in (Ochoa and Cortesão, 2019; Nemec et al., 2018), which does not fully exploit the benefits of introducing the human user in the task, mainly his or her adaptability to unforeseen changes in the conditions of the workspace or the target object.

In order to get more from the human user's skills, a higher degree of interaction can be accomplished by including him or her in the actual online task. Nevertheless, even if the human and the robot system cooperate by working on the same target object and being present in the same workspace, some previous research (Nemec et al., 2018; Gaz et al., 2018) do not actually give the human the ability to command the operation carried out by the robot by interacting with it, so even though the human user's skills are present in the online task, they are not fully grasped, as his or her initiative is not synergistically combined with the accuracy and strength of the robot system.

This question is addressed in work (Gracia et al., 2019), where the human user guides the robot to perform the treatment on a specific area. Even though this is an example of a truly cooperative application, the degree of interaction can still be deepened by further exploiting the robot system part, including the possibility for the robot to combine an automated task with the manual guiding by the user and developing control laws which ensure the correct completion of the operation.

2.1.2 Assisted teleoperation

Teleoperation, i.e. the remote control of a robot by a human operator, is one of the first manifestations of robotics (Niemeyer et al., 2016) and is still an on-

going tendency in research. Its fields of interest are wide, including situations where the environment is either unsafe for humans, such as operations in space (Chen et al., 2019) and in radioactive areas (Bandala et al., 2019; Abi-Farraj et al., 2020), or difficult to reach, such as aerial (Suarez et al., 2020; Isop et al., 2019) and subaquatic

However, a rich body of contributions in the field of teleoperation, of which this work is part of, has been developing, with its focus put on deepening and improving human-robot interaction in teleoperation (Selvaggio et al., 2018; Nicolis et al., 2018; Lu et al., 2018; Girbés-Juan et al., 2021; Gorjup et al., 2019; Clark et al., 2019), rather than solving specific problems such as the ones cited above.

Regarding the degree of shared-control between the human operator and the robotic system, two extremes of a spectre might be identified (Niemeyer et al., 2016), one being direct control, where the robotic systems limits itself to the direct execution of the human's commands, and the other being supervisory control, where the robotic system carries out highly automated tasks, and the human operator makes high-level decisions.

Still in the field of direct control, but with a more assisted approach of teleoperation, telepresence (Niemeyer et al., 2016) provides the human operator with an interface which makes the direct control task less dependent on his or her skills and concentration.

Although this is a necessary approach in some fields, especially in surgery, where non-vision-based reactive control methods (such as force and impedance control) are hard to combine with a non invasive approach (small sensors, soft tissues) (Lopez et al., 2013), assisted teleoperation with telepresence interfaces and virtual barriers present inconveniences due to the fact that while control is still mainly carried out by the human operator, his or her maneuvers are limited by passive restrictions which are not evident in a perceptual sense, (Selvaggio et al., 2018).

As a result, shared-control architectures where the robotic system plays an active role in the task, while preserving the initiative of the human operator, are being addressed in various contributions. Whereas some of them are based on trajectory planners, such as (Lv et al., 2020), this offers some limitations regarding computational cost and real-time control in comparison to reactive control methods, such as the case of this work, among others (Selvaggio et al., 2018; Suarez et al., 2020).

2.1.3 Augmented Reality-based Interfaces

Human-machine interfaces are devices that allow the interaction between a human and a machine (Tonin and Millan, 2021; Tang et al., 2021). If the interface is placed inside the brain or body of the human, it is known as an invasive or implanted interface (Jin et al., 2019). On the contrary, if the interface is external to the human body, it is known as a non-invasive or wearable interface (Dumitrescu et al., 2021; Xu et al., 2022; Cao et al., 2021). Chapter 6 is focused on non-invasive interfaces and on how to develop this kind of interfaces for complex robotic applications.

Technological advances in the creation of holograms have nowadays made it possible to have devices and software tools that allow Augmented Reality (AR) applications in industrial sectors (Muñoz et al., 2019, 2020; Chu and Chang, 2021; Tedesco et al., 2021). In short, augmented reality projects holograms into physical space, allowing for a more intuitive and natural interaction between human and machine (Craig, 2013).

Some previous works used AR interfaces to improve robot teleoperation for industrial tasks, such as (Li et al., 2019; Rosen et al., 2019; Gadre et al., 2019; Mistry and Maes, 2009; Ismail et al., 2019), which are discussed in detail in Chapter 6.

Although this is assessed thoroughly in Chapter 6, most of the AR approaches mentioned above developed solutions for robot-object manipulation tasks, whereas this work proposes a new AR interface for industrial complex tasks, such as surface treatment tasks involving a bimanual robot system.

2.2 Surface treatment

Surface treatment presents two phases with different requirements to be addressed:

- **Contact phase:** This is the phase where the surface treatment actually takes place. This phase requires continued and relatively homogeneous contact combined with orthogonality between the workpiece surface and the robot tool (Perez-Vidal et al., 2019). Some aditional, non-esential requirements, such as always remaining on the workpiece surface or being able to addapt to changes in the workpiece (p.e. its position or orientation) must be taken into account).

– **Non-Contact phase:** This is the phase previous to the operation itself, where the requirements are: locating the workpiece, approaching it smoothly and ensuring that the first contact between the robot tool and the surface is orthogonal.

Some *tools* have been specifically designed to properly accomplish surface treatment tasks using robot systems (Fer, 0610; Kuo et al., 2019; Mohammad et al., 2018).

On the other hand, the complete or partial automation of surface treatment tasks, such as polishing, sanding or deburring, has been the main focus of a growing number of research contributions in the last years. For instance, some works addressed the automation of surface treatment using robotic systems (Solanes et al., 2019; Yuan et al., 2019) or human-robot collaboration to solve this issue (Gracia et al., 2019), whereas other works tackled some specific issues, such as detecting wether a polishing operation is complete or the polishing tool needs to be changed (Segreto and Teti, 2019).

The shortcomings of this previous research will be discussed in detail in Chapter4, but it must be noted that none of them tackles all of the requirements present both in the contact and non-contact phase, as this work does.

2.3 Bimanual robotics

The introduction of dual-arm robotic systems in industrial, domestic and assistive tasks is justified by their dexterity, flexibility, manipulability and their general resemblance to human behaviour when it comes to solving tasks, which makes it easier for human operators to relate to them and to design human-like applications more intuitively (Smith et al., 2012). As a result of sustained remarkable interest in dual-arm robotic systems from industry and academia (Makris et al., 2017), a wide variety of contributions in this field is available.

Some of these contributions can be described as goal-coordinated dual-arm robotic applications, where two robot manipulators work on the same task without physical interaction between them (Smith et al., 2012): for instance, in (Selvaggio et al., 2018)

However, a main trend in dual-arm robotics, including [CHAPTER BIMANUAL] in this work, focuses on bimanual robotics, which consists of dual-arm robotic systems where the two robot manipulators are coordinated to achieve a shared goal and interact physically in order to do so (Smith et al.,

2012).

A significant number of bimanual robotics contributions develop fully automated applications, (Joshi et al., 2020; García et al., 2019; Mitash et al., 2020; Sintov et al., 2020; McConachie et al., 2020; Zimmermann et al., 2020; Liang et al., 2020)

Nonetheless, Human-Robot Interaction is specially interesting for bimanual robotics not only because of the general reasons previously explained, but also because it exploits the human's intuitive understanding of bimanual configurations and movements, and also, because the methods used to implement Human-Robot Interaction are extended and applied to the cooperation between robots (both physically and at control level), which allows the human-user to interact with more complex robot systems on the same basis as in single robot configurations.

2.4 Task optimization

In order to accomplish the robot task, some kind of optimization has to be considered when developing the control algorithm for the robot system. These optimization problems are well known in robotics and have been typically approached in two different ways, broadly classified as planning methods and "reactive" controllers (Siciliano and Khatib, 2008), as discussed below.

On the one hand, robot controllers based on high-level planning mainly solve the optimization problem taking into account the complete data of the robot task. This type of approach is suitable to cope with trap situations (Gracia et al., 2012) and singular configurations (Gracia et al., 2009), but it typically suffers from high computational cost and the difficulty to deal with task uncertainty. Examples of this type of approach can be found in (Da Silva et al., 2020; Li et al., 2019).

On the other hand, robot controllers based on reactive algorithms solve the optimization problem without prediction capabilities, i.e., considering only the data associated to the current time instant in order to compute the current control action to be applied to the robot system. This type of approach can be readily used in real-time to control the robot system, although it may suffer from trap situations and singular configurations. Examples of this type of approach are the navigation algorithms based on the well known artificial potential-fields (Park et al., 2020; Li and Li, 2020) or, more recently, based on neural networks (Khan et al., 2020) and SMC (Khan and Li, 2020; Fei et al.,

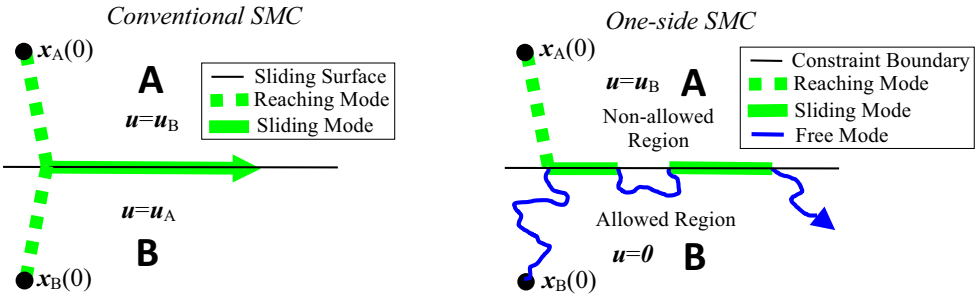


Figure 2.1: Graphical comparison between conventional SMC (left) and one-side SMC (right).

2020), among others.

The method developed in this work belongs to this category of reactive algorithms. In particular, several controllers based on SMC theory (Su and Zheng, 2020; Zhang et al., 2020) are developed in this work in order to benefit from its inherent advantages, such as robustness and low computational cost. Moreover, a task prioritization strategy is also used in this work to simultaneously address a set of objectives in order to properly perform surface treatment tasks. Thus, the objectives are associated to a set of equalities whose square errors are hierarchically minimized according to the priority assigned to each objective/equality.

2.5 Sliding Mode Control

Sliding Mode Control (SMC) Utkin et al. (2009) is widely utilized in robot tasks since it has the inherent advantages of robustness and low computation cost, e.g., see (Baek et al., 2016; Lee et al., 2017), among others. In this section, the basic concepts for SMC are presented:

2.5.1 Conventional SMC

For conventional SMC, see Fig. 2.1-left, the state space of the system is divided into two regions, **A** and **B**, separated by the sliding surface.

When the system is in **A** the control action $\mathbf{u} = \mathbf{u}_B$ "pushes" the system into **B**. Similarly, the control action $\mathbf{u} = \mathbf{u}_A$ "pushes" the system into **A** when

the system is in **B**.

Thus, in both cases the system evolves to the sliding surface, which is called *reaching mode* (Utkin et al., 2009). Then, the control action **u** switches between $\mathbf{u}_A$ and $\mathbf{u}_B$ at a theoretically infinite frequency to keep the system on the sliding surface, which is known as *sliding mode* (SM) (Utkin et al., 2009).

Moreover, there is a *continuous equivalent control* (Edwards and Spurgeon, 1998) for the SM phase that keeps the system on the sliding surface, although SMC produces such control action without explicitly computing it and with low computational cost (Edwards and Spurgeon, 1998).

2.5.2 Non-Conventional SMC

This work proposes the one-side SMC depicted in Fig. 2.1-right, which is useful to satisfy inequality constraints.

For this approach, the system state space is divided into the allowed region **B** and non-allowed region **A**, which are separated by the constraint boundary.

As before, the control action $\mathbf{u} = \mathbf{u}_B$ pushes the system into **B** when the system sate is in **A**. Nevertheless, no control action is applied ($\mathbf{u} = \mathbf{0}$) when the system state is in **B**.

Thus, the system evolves in reaching mode to the sliding surface when it starts in **A**, although the system state can "freely" evolve according to some other criterion when it starts in **B**.

Hence, only when the state trajectory tries by itself to leave the allowed region, the one-side SMC will make **u** switch between **0** and $\mathbf{u}_B$ at a theoretically infinite frequency, which can be seen as an ideal SM behaviour (Edwards and Spurgeon, 1998).

2.6 Computer vision

Computer vision is widely used in industrial robot tasks since it provides flexibility and precision. The camera can be placed on the end-effector of the robot system (eye-in-hand configuration), e.g., see (Cui et al., 2020), or, alternatively, it can be placed on a structure to "observe" the whole workspace of the robot system (eye-to-hand configuration), e.g., see (Taryudi and Wang, 2018).

The visual data can be obtained using 2D cameras, estimating depth by processing the images (Nilsen et al., 2019; Muñoz-Benavent et al., 2019; As-

tanin et al., 2017), or can be directly read from 3D sensors (Ferraro et al., 2012), such as Microsoft Kinect.

Although all of these solutions are valid for certain applications, an eye-to-hand camera network presents important advantages for the surface treatment process: the workpiece location is obtained globally, i.e., it is not constrained by the robot kinematics (contrarily to the eye-in-hand configuration), and, hence, larger areas may be covered; and the camera network represents a *redundant* system that provides accuracy and robustness, e.g., when the robot or human operator occlude the field of view of a camera, the remaining cameras are able to properly locate the workpiece.

Registration is the process of getting the transformation needed to express two views of an object under the same coordinate system (Bi and Wang, 2010). This process is useful to obtain the position and orientation (i.e., the pose) of a workpiece by matching the *point cloud* acquired by the camera with the virtual model of the workpiece. In particular, Chapter 4 of this work uses the Iterative Closest Point (ICP) method (He et al., 2017) to obtain the pose of the workpiece since it is widely used due to its effectiveness and simplicity.

In the next chapter, the theoretical preliminaries and the principles of sliding mode control are reviewed.

Chapter 3

Theoretical Basis

This chapter reviews the theoretical basis used to develop the proposed approach in Chapters 4, 5 and 6, with a special focus on reviewing the principles of Sliding Mode Control.

3.1 Preliminaries

3.1.1 Kinematics

The kinematics of a robot system can be expressed as:

$$\mathbf{p} = \mathbf{l}(\mathbf{q}) \tag{3.1}$$

$$\dot{\mathbf{p}} = \frac{\partial \mathbf{l}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{3.2}$$

$$\ddot{\mathbf{p}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \tag{3.3}$$

being $\mathbf{p} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^{\mathrm{T}}$ the pose of the robotic system, where α, β and γ represent the orientation angles (roll, pitch and yaw, respectively), $\mathbf{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^{\mathrm{T}}$ the configuration of the robotic system, $\mathbf{J}$ the Jacobian matrix and $\mathbf{l}$ the so-called kinematic function (Chiaverini et al., 2008).

3.1.2 Task-priority based strategy

This method is useful to address a set of tasks with different priorities, where the error of the task equations has to be minimized. The recursive equations of this strategy are given below (Nakamura et al., 1987):

$$\mathbf{A}_i \overline{\mathbf{x}} = \mathbf{b}_i, \qquad i = 1, \ldots, M, \tag{3.4}$$

$$\overline{\mathbf{x}}_i = \overline{\mathbf{x}}_{i-1} + (\mathbf{A}_i \mathbf{N}_{i-1})^{\dagger} (\mathbf{b}_i - \mathbf{A}_i \overline{\mathbf{x}}_{i-1}), \qquad i = 1, \ldots, M, \tag{3.5}$$

$$\mathbf{N}_i = \mathbf{N}_{i-1} (\mathbf{I} - (\mathbf{A}_i \mathbf{N}_{i-1})^{\dagger} (\mathbf{A}_i \mathbf{N}_{i-1})), \qquad i = 1, \ldots, M, \tag{3.6}$$

being M the considered number of equalites or tasks, $\overline{\mathbf{x}}$ is the unknown vector to be computed (which, in this work, corresponds to the commanded joint accelerations $\ddot{\mathbf{q}}_c$), $\mathbf{A}_i$ and $\mathbf{b}_i$ are the matrix and vector, respectively, for the i-th task ($i = 1$ represents the highest priority), $\overline{\mathbf{x}}_i$ is the solution that hierarchically minimizes the error of the first i task equations, $\mathbf{N}_0 = \mathbf{I}$ is the identity matrix, $\overline{\mathbf{x}}_0 = \mathbf{0}$ is the null vector and superscript $\dagger$ denotes the pseudoinverse of a matrix (Golub and Van Loan, 1996) (a threshold can be used to neglect the small singular values).

Note that, in this work, the unknown vector $\overline{\mathbf{x}}$ to be computed corresponds to the commanded joint accelerations, i.e., $\ddot{\mathbf{q}}_c$. Moreover, $\overline{\mathbf{x}}_M$ denotes the solution to this unknown vector for the M task equations, which is computed

by the above task prioritization algorithm. Basically, this algorithm hierarchically minimizes the Least Square Error (LSE) of the task equations, i.e., a lower-priority task is satisfied only by using the degrees of freedom in the null space of the higher-priority ones. See (Nakamura et al., 1987) for further details.

3.1.3 Boundary model

The first levels of the control algorithms in Chapters 4 and 5 include inequality constraints required to keep the center of the workpiece and the surface treatment tool within the allowed workspace, whose boundary has the shape of a superellipsoid in Chapter 4 and one of the control algorithms in Chapter 5 and a modified superellipse in the second control algorithm in Chapter 5.

The equation that defines a superellipsoid is given by:

$$\left|\frac{x}{W}\right|^m + \left|\frac{y}{H}\right|^m + \left|\frac{z}{M}\right|^m = 1, \tag{3.7}$$

where $\{W, H, M\}$ are the superellipsoid axes and exponent m, whose value has to be greater than one, defines the smoothing of the superellipsoid. That is, the shape of the superellipsoid ranges from an ellipsoid to a rectangular cuboid as m ranges from 2 to infinity. In this work, the chosen value for m is 4.

In the case of a superellipse, the third dimension is removed, while maintaining the same exponent m. Therefore, the equation for the superellipse, which resembles a $2W \times 2H$ rectangle with smoothed corners, is given by:

$$\left|\frac{x}{W}\right|^m + \left|\frac{y}{H}\right|^m = 1. \tag{3.8}$$

However, in this work a modified superellipse is used instead of the one described above in order to obtain a more homogeneous control action to keep the robot tool within the allowed workspace when one side of the superellipse is significantly longer than the other, as it is further explained in Section 5.2.6.1.

The equation that describes the proposed modified superellipse is:

$$\left|\frac{x}{W}\right|^m + \left(\frac{\max(|y| - (H - W), 0)}{W}\right)^m = 1, \tag{3.9}$$

where it has been assumed $H > W$, although it can be readily modified for the analogous case $H < W$, details omitted for brevity.

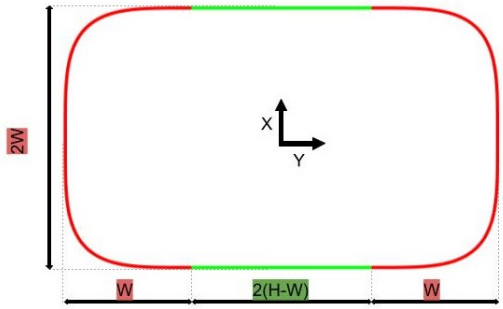


Figure 3.1: Modified superellipse proposed in this work, which is composed of a $2W \times 2(H-W)$ rectangle and two offseted halfs of an even-sided $2W \times 2W$ superellipse.

As shown in Fig. 3.1, the equation above describes a rectangle with smoothed corners, with $2W$ for its short side and $2H$ for its long side, by attaching a $2W \times 2(H-W)$ rectangle to two offseted halfs of an even-sided $2W \times 2W$ superellipse.

3.2 Sliding Mode Control

This section further explains the SMC algorithms introduced in Chapter 2: on the one hand, a conventional SMC used to satisfy equality constraints and, on the other hand, a novel one-side SMC proposed to satisfy inequality constraints.

3.2.1 Conventional SMC to satisfy equality constraints

As introduced in Chapter 2, a conventional SMC is developed in the theorem below to fulfill equality constraints.

Theorem 3.2.1. *Consider the following dynamical system with n_x states and n_u inputs given by:*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{d}) + \mathbf{g}(\mathbf{x})\,\mathbf{u}, \tag{3.10}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{d}(t)$ is an unmeasured disturbance or model uncertainty, $\mathbf{u}(t)$ is the control input vector (possibly discontinuous), $\mathbf{f}$ is a vector field and $\mathbf{g}$ is a set of vector fields.

Consider also that the system state vector $\mathbf{x}$ *is subject to* equality *constraints* $\phi_{eq,i}(\mathbf{x}) = 0$, $i = 1, \ldots, N_{eq}$, *where* $\phi_{eq,i}(\mathbf{x})$ *is the* i*th equality constraint function. Thus, the region* Φ_{eq} *of the state space compatible with the constraints on state* $\mathbf{x}$ *is given by:*

$$\Phi_{eq} = \{\mathbf{x} \mid \phi_{eq,i}(\mathbf{x}) = 0\}, \tag{3.11}$$

with $i = 1, \ldots, N_{eq}$.

Then, assuming that the constraint functions $\phi_{eq,i}$ *are differentiable, the control action* $\mathbf{u}$ *that satisfies the control law below guarantees the convergence of the system to* Φ_{eq} *in finite time and remains there henceforth:*

$$\mathbf{L_g}\boldsymbol{\phi}_{eq}\mathbf{u} = -\mathbf{W}_{eq} sign(\boldsymbol{\phi}_{eq})\, u_{eq}^{+} \tag{3.12}$$

$$u_{eq}^{+} > \|L_f\boldsymbol{\phi}_{eq}\|_1 \Big/ diag_{\min}(\mathbf{W}_{eq}), \tag{3.13}$$

where column vector $\boldsymbol{\phi}_{eq}$ *is composed of all the functions* $\phi_{eq,i}$, *the scalar* $L_f\phi_{eq,i} = \frac{\partial \phi_{eq,i}^T}{\partial \mathbf{x}}\mathbf{f}$ *and the row vector* $\mathbf{L_g}\phi_{eq,i} = \frac{\partial \phi_{eq,i}^T}{\partial \mathbf{x}}\mathbf{g}$ *represent the Lie derivatives of* $\phi_{eq,i}(\mathbf{x})$ *in the direction of* $\mathbf{f}$ *and* $\mathbf{g}$, *respectively,* $L_f\boldsymbol{\phi}_{eq}$ *is a column vector composed of the elements* $L_f\phi_{eq,i}$ *of all equality constraints,* $\mathbf{L_g}\boldsymbol{\phi}_{eq}$ *is a matrix composed of the row vectors* $\mathbf{L_g}\phi_{eq,i}$ *of all equality constraints,* $sign(\cdot)$ *denotes the* sign function, u_{eq}^{+} *is a positive scalar representing the switching gain, diagonal matrix* $\mathbf{W}_{eq}$ *is composed of the constraint switching gain weights,* $\|\cdot\|_1$ *denotes the Taxicab or 1-norm and* $diag_{\min}(\cdot)$ *returns the minimum value of the diagonal elements of a matrix.*

Proof. The proof is similar to the generalization of Proof 2.1 in (Utkin et al., 2009). See (Utkin et al., 2009) for further details. □

3.2.2 One-side SMC to satisfy inequality constraints

Using the approach exposed in Chapter 2, the theorem below is developed to fulfill inequality constraints.

Theorem 3.2.2. *Consider the dynamical system given by* (3.10) *and consider also that the system state vector* $\mathbf{x}$ *is subject to* inequality *constraints* $\phi_{in,i}(\mathbf{x}) \leq 0$, $i = 1, \ldots, N_{in}$, *where* $\phi_{in,i}(\mathbf{x})$ *is the* i*th inequality constraint function. Thus, the region* Φ_{in} *of the state space compatible with the constraints on state* $\mathbf{x}$ *is given by:*

$$\Phi_{in} = \{\mathbf{x} \mid \phi_{in,i}(\mathbf{x}) \leq 0\}, \tag{3.14}$$

with $i = 1, \ldots, N_{in}$.

Then, assuming that the constraint functions $\phi_{in,i}$ *are differentiable, the control action* $\mathbf{u}$ *that satisfy the control law below guarantees the convergence of the system to* Φ_{in} *in finite time and remains there henceforth:*

$$v2dm\left(pos\left(\boldsymbol{\phi}_{in}\right)\right)\mathbf{L_g}\boldsymbol{\phi}_{in}\mathbf{u} = -\mathbf{W}_{in}\, pos\left(\boldsymbol{\phi}_{in}\right)\, u_{in}^{+} \tag{3.15}$$

$$u_{in}^{+} > \sum_{i=1}^{n_a} \left(\max(L_f\phi_{in,i}, 0)\right) / diag_{\min}(\mathbf{W}_{in}), \tag{3.16}$$

where $v2dm(\cdot)$ *converts a vector into a diagonal matrix,* $pos(\cdot)$ *denotes the* positive function *(i.e.,* $pos(x)$ *is equal to* 0 *if* $x < 0$ *and equal to* 1 *if* $x > 0$*), function* $\max(a_1, \ldots, a_i)$ *returns the maximum value out of the arguments* $a1, \ldots, a_i$*, column vector* $\boldsymbol{\phi}_{in}$ *is composed of all the functions* $\phi_{in,i}$*,* $\mathbf{L_g}\boldsymbol{\phi}_{in}$ *is a matrix composed of the row vectors* $\mathbf{L_g}\phi_{in,i}$ *of all inequality constraints, the scalar* $L_f\phi_{in,i} = \frac{\partial \phi_{in,i}^T}{\partial \mathbf{x}}\mathbf{f}$ *and the row vector* $\mathbf{L_g}\phi_{in,i} = \frac{\partial \phi_{in,i}^T}{\partial \mathbf{x}}\mathbf{g}$ *represent the Lie derivatives of the inequality constraints in the direction of* $\mathbf{f}$ *and* $\mathbf{g}$*, respectively,* u_{in}^{+} *is a positive scalar representing the switching gain, diagonal matrix* $\mathbf{W}_{in}$ *is composed of the constraint switching gain weights and* n_a *denotes the number of active inequality constraints, i.e., those with* $\phi_{in,i} \geq 0$.

It is worth noting that $\text{v2dm}(\text{pos}(\boldsymbol{\phi}_{in}))$ on the left-side of (3.15) yields the trivial scalar equation $0 = 0$ for the non-active inequality constraints (i.e., those with $\phi_{in,i} < 0$) and, thus, these constraints do not use system DoF.

Proof. First, the inequality constraint vector is partitioned into two subvectors $\boldsymbol{\phi}_{in} = [\boldsymbol{\phi}_{in}^{n_a\,\mathrm{T}} \quad \boldsymbol{\phi}_{in}^{N_{in}-n_a\,\mathrm{T}}]^{\mathrm{T}}$, where the first subvector is composed of the n_a active inequality constraints (i.e., those with $\phi_{in,i} \geq 0$) and the second subvector of the remaining non-active inequality constraints (i.e., those with $\phi_{in,i} < 0$).

Assuming[1] that $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$, the objective of this proof is to show that convergence to point $\boldsymbol{\phi}_{in}^{n_a} = \mathbf{0}$ is achieved in finite time.

[1] Note that it has been assumed, without loss of generality, that the active inequality constraints do not initially fulfill the constraints, i.e., $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$, since otherwise the controlled system simply remains at the starting point $\boldsymbol{\phi}_{in}^{n_a}(0) = \mathbf{0}$, which is the goal of the SMC. Therefore, $\boldsymbol{\phi}_{in}^{n_a}(0) > \mathbf{0}$ is the *general case* for the starting point and does not pose any requirement for the task of the robotic system.

The vector $\dot{\boldsymbol{\phi}}_{in}$ composed of the function derivatives $\dot{\phi}_{in,i}$ is given by

$$\dot{\boldsymbol{\phi}}_{in} = \frac{\partial \boldsymbol{\phi}^{\mathrm{T}}}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{d}) + \frac{\partial \boldsymbol{\phi}^{\mathrm{T}}}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})\, \mathbf{u} = L_f \boldsymbol{\phi}_{in} + \mathbf{L_g} \boldsymbol{\phi}_{in} \mathbf{u}. \tag{3.17}$$

Premultiplying (3.17) by $\mathrm{v2dm}\,(\mathrm{pos}\,(\boldsymbol{\phi}_{in}))$ and substituting (3.15) yields:

$$\mathrm{v2dm}(\mathbf{z}_{in}) \dot{\boldsymbol{\phi}}_{in} = \mathrm{v2dm}(\mathbf{z}_{in}) L_f \boldsymbol{\phi}_{in} - \mathbf{W}_{in}\, \mathbf{z}_{in}\, u_{in}^{+}, \tag{3.18}$$

where column vector $\mathbf{z}_{in}$ has the ith-component $z_{in,i} = 1$ if $\phi_{in,i} > 0$ and $z_{in,i} = 0$ if $\phi_{in,i} < 0$.

Let $V_{in} = \mathbf{z}_{in}^{\mathrm{T}}\, \mathrm{v2dm}(\mathbf{z}_{in})\, \boldsymbol{\phi}_{in}$ be a Lyapunov function candidate. Vector $\boldsymbol{\phi}_{in}^{n_a}$ can be generically partitioned into two subvectors $\boldsymbol{\phi}_{in}^{n_a} = [\boldsymbol{\phi}_{in}^{b\,\mathrm{T}} \quad \boldsymbol{\phi}_{in}^{n_a-b\,\mathrm{T}}]^{\mathrm{T}}$, where SM occurs in the manifold given by $\boldsymbol{\phi}_{in}^{b} = \mathbf{0}$, whereas the components of vector $\boldsymbol{\phi}_{in}^{n_a-b}$ are greater than zero. Note that one of these two subvectors may be empty at a certain time. Taking into account that $\mathbf{z}_{in}^{n_a-b} = \mathbf{1}$ and $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$, the Lyapunov function derivative is given by:

$$\begin{aligned} \dot{V}_{in} = \frac{d}{dt} \left(\begin{bmatrix} \mathbf{z}_{in}^{b} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}^{\mathrm{T}} \mathrm{v2dm} \left(\begin{bmatrix} \mathbf{z}_{in}^{b} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \right) \right) \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\phi}_{in}^{n_a-b} \\ \boldsymbol{\phi}_{in}^{N_{in}-n_a} \end{bmatrix} + \mathbf{z}_{in}^{\mathrm{T}}\, \mathrm{v2dm}(\mathbf{z}_{in})\, \dot{\boldsymbol{\phi}}_{in} \\ = \mathbf{z}_{in}^{\mathrm{T}}\, \mathrm{v2dm}(\mathbf{z}_{in})\, \dot{\boldsymbol{\phi}}_{in}. \end{aligned} \tag{3.19}$$

Substituting (3.18) in (3.19) yields:

$$\dot{V}_{in} = \mathbf{z}_{in}^{\mathrm{T}}\, \mathrm{v2dm}(\mathbf{z}_{in})\, L_f \boldsymbol{\phi}_{in} - \mathbf{z}_{in}^{\mathrm{T}}\, \mathbf{W}_{in}\, \mathbf{z}_{in}\, u_{in}^{+}. \tag{3.20}$$

Since $\mathbf{z}_{in}^{N_{in}-n_a} = \mathbf{0}$ and the components of vector $\mathbf{z}_{in}^{n_a}$ range from 0 to 1, the upper bound of the first term in (3.20) is given by $z_{in,i}^{n_a} = 1$ if $L_f \phi_{in,i}^{n_a} > 0$ and $z_{in,i}^{n_a} = 0$ if $L_f \phi_{in,i}^{n_a} < 0$, that is:

$$\mathbf{z}_{in}^{\mathrm{T}}\, \mathrm{v2dm}(\mathbf{z}_{in})\, L_f \boldsymbol{\phi}_{in} \leq \sum_{i=1}^{n_a} (\max(L_f \phi_{in,i}, 0)). \tag{3.21}$$

Assuming[2] that $u_{in}^{+} > 0$, the second term in (3.20) is negative, since matrix

[2]Note that the switching gain u_{in}^{+} is a free design parameter that has been assumed to be positive in order to apply the control action of the SMC in the proper direction. This is analogous, for instance, to the case of the classical "proportional controller" where, for stability reasons, the correction gain is assumed to be positive in order to apply the corrective control action in the direction of the error signal.

$\mathbf{W}_{in}$ is positive definite, and its upper bound is given by:

$$-\mathbf{z}_{in}^{\mathrm{T}}\,\mathbf{W}_{in}\,\mathbf{z}_{in}\,u_{in}^{+} \leq -\mathrm{diag}_{\min}(\mathbf{W}_{in})\,\|\mathbf{z}_{in}\|_2^2\;u_{in}^{+}, \quad \text{where } \|\mathbf{z}_{in}\|_2 \geq 1\;\forall\,\boldsymbol{\phi}_{in} > \mathbf{0}, \tag{3.22}$$

because if vector $\boldsymbol{\phi}_{in}^{n_a-b}$ is not empty at least one component of vector $\mathbf{z}_{in}$ is equal to 1.

Taking into account (3.21) and (3.22), the Lyapunov function derivative has the following upper bound:

$$\dot{V}_{in} \leq \sum_{i=1}^{n_a} (\max(L_f\phi_{in,i}, 0)) - \mathrm{diag}_{\min}(\mathbf{W}_{in})\,u_{in}^{+}. \tag{3.23}$$

Therefore, if u_{in}^{+} fulfills (3.16) the right-hand of Eq. (3.23) is negative and, hence, the derivative of the Lyapunov function (i.e., the left-hand of Eq. (3.23)) is negative. Thus, the Lyapunov function decreases at a finite rate, it vanishes and SM occurs in the intersection of the active inequality constraints. Note that this conclusion is analogous to that in the proof of conventional SMC in (Utkin et al., 2009). □

3.2.3 Modified constraints

The original constraints $\sigma_{eq,i}$ and $\sigma_{in,i}$ are modified as follows to include also the velocity of the robotic system:

$$\phi_{eq,i} = \sigma_{eq,i} + K_{eq,i}\dot{\sigma}_{eq,i} = 0 \tag{3.24}$$

$$\phi_{in,i} = \sigma_{in,i} + K_{in,i}\dot{\sigma}_{in,i} \leq 0, \tag{3.25}$$

where design parameters $K_{eq,i}$ and $K_{in,i}$ establish the approaching speed to the equality constraint manifold and to the boundary of the inequality constraint, respectively.

In the next chapter, an approach based on sliding-mode ideas is proposed to satisfy different types of constraints in visual servoing.

Chapter 4

Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints

4.1 Introduction

4.1.1 Objective

As it has been already stablished in Chapters 1 and 2, whereas there have been several solutions developed to solve the detection of surface anomalies, the repairing part of the process is still mainly performed by human operators. Thus, this work aims to automate surface treatment operations by means of a synergistic cooperation between a robot system and the human operator.

In particular, in the application developed in this work the human operator provides flexibility, guiding the tool of the robot system to treat arbitrary regions of the workpiece surface; while the robot system provides strength, accuracy and security, not only holding the tool and keeping the right tool orientation, but also guaranteeing a smooth approach to the workpiece and confining the tool within the allowed area close to the workpiece. Moreover, to

add more flexibility to the proposed method, when the user is not guiding the robot tool, a robot automatic operation is activated to perform the treatment in prior established regions. Furthermore, a camera network is used to get a global view of the robot workspace in order to obtain the workpiece location accurately and in real-time. The effectiveness of the proposed approach is shown with several experiments using a 6R robotic arm.

It must be noted that this chapter corresponds to two published contributions: (García et al., 2020) and (García et al., 2021). Where (García et al., 2021) contributes with a expansion of (García et al., 2020)] with the addition of a boundary constraint to the control scheme. This being the case, this chapter is mostly based on (García et al., 2021), although the results from both applications are shown in the corresponding section.

4.1.2 State of the Art

4.1.2.1 Robot tool

As addressed in Chapter 2, several manufacturers and reasearchers have designed tools specifically for automated surface treatment tasks:

For instance, the manufacturer FerRobotics has a product line of robot tools, named *active contact flange* (Fer, 0610), which basically consist of a pneumatic system that allows the robot to reliably meet the force required to properly accomplish the surface treatment task.

Similarly to the previous approach, an adjustable force regulation mechanism to be placed in the end-effector of the robotic system was designed in (Kuo et al., 2019) to control the contact force for grinding and deburring operations without using a force sensor.

Moreover, the conventional structure of the robot end-effector used for polishing tasks was redesigned in (Mohammad et al., 2018) to obtain a low-inertia effect in order to reduce vibrations and to improve the tracking of the required contact force.

4.1.2.2 Automatic mode of operation

As exposed in Chapter 2, most of the research regarding surface treatment tries to find completely automated solutions, with no interaction with the human operator. Some representative examples are discussed below:

In (Villagrossi et al., 2017), a laser sensor mounted on the robot end-

effector was used to previously scan the target workpiece in order to generate the trajectories that were subsequently followed by the robot system to perform the deburring operation. Similarly to the previous approach, a vision system mounted on the robot system was used in (Diao et al., 2018) to previously position the machining target, which was subsequently treated by the grinding robot.

In (Mohsin et al., 2017; Mohsin et al., 2019), a tool path planning algorithm with controlled force and polishing parameters optimization was developed to perform a polishing operation using a robot arm that held the workpiece that was being polished by an external passive polishing tool.

In (Oba and Kakinuma, 2017), a serial-parallel robot was considered to polish unknown curved surfaces using data provided by a force sensor to properly adapt the tool posture and the polishing pressure. Similarly, in (Tian et al., 2016) a polishing pressure model was considered to improve the polishing quality on curved surfaces by ensuring a constant polishing pressure.

In (Sato et al., 2017), a force-sensorless control was developed to grind an object with a robot system using the analytic relation between the contact force and the grinding resistance.

In (Segreto and Teti, 2019), a machine learning algorithm based on multiple sensor data was developed to determine the state of the robot-assisted polishing process in order to establish the end-point of the polishing process.

In (Huang et al., 2020), a force planning strategy based on a compliance model was developed to avoid large instantaneous contact forces when the grinding tool of the robot contacted the workpiece. Moreover, an algorithm for automatic contour surface processing and target force tracking was proposed in order to simplify the programming of the grinding task.

In (Kharidege et al., 2017), a path planning application was developed based on the CAD data of the workpiece to be polished. Note that this is a basic approach where the reference points for the robot system were computed offline and no kind of pressure adaptation was performed while the robot was polishing the workpiece.

4.1.2.3 Human-robot Cooperation

Some previous research considered some sort of interactions between the human operator and the robot system when performing the surface treatment tasks.

In (Ochoa and Cortesão, 2019), impedance control was used to program a polishing task using the teaching by demonstration method. That is, firstly the human operator guided the robot tool by hand in order to "teach" the robot the polishing pattern of positions and forces. Subsequently, the robot automatically performed the polishing task tracking the mentioned pattern of positions and forces. Note that, in this approach, the interactions between the operator and the robot system were only considered at a previous stage, i.e., to program the polishing task *offline*. Moreover, note that the robot system was not able to adapt to unforeseen changes of the target object, e.g., changes in position, shape, size, consistency, etc.

Similarly to the previous approach, in (Nemec et al., 2018) the teaching by demonstration method was used to initially register the position and force data measured from a skilled operator. Subsequently, the robot system held the workpiece while it was being polished by an external passive tool. For this purpose, the virtual mechanism approach was used to characterize the closed kinematic chain composed by the union of the robot arm and the external polishing tool.

Furthermore, in (Gaz et al., 2018) a human-robot collaboration application for manual polishing was developed, where the robot arm held the workpiece and the operator used an abrasive tool to perform the polishing task. During the operation the robot arm maintained the workpiece in a fixed position and the operator was able to modify its orientation by pushing the robot body, which was detected by a force sensor attached to the wrist of the robot. However, it is worth noting that this was a "passive" robotic application, since the human user had to treat the surface with the tool, keeping both the desired pressure and the orthogonality between the tool of the robot system and the workpiece surface.

The application of surface treatments requires not only contact but also orthogonality between the workpiece surface and the robot tool (Perez-Vidal et al., 2019). For instance, a force/torque sensor mounted on the robot end-effector was utilized in (Gracia et al., 2019) to simultaneously regulate the contact pressure and the mentioned perpendicularity. Furthermore, a second force sensor attached to the robot tool was used in (Gracia et al., 2019) to allow the operator guiding the tool to treat an arbitrary region of the workpiece surface.

In (Santos and Cortesão, 2018), a teleoperation system was developed in order to conduct echographies using a medical robot, i.e., a physician teleoper-

ated, using a haptic device, a robot equipped with an ultrasound probe in order to obtain ultrasound images from a patient. In this application the physician perceived, by means of the haptic device, the stiffness of the patient, which was detected by means of a force sensor mounted on the robot end-effector. Moreover, a virtual progressive stiffness before contact was considered to facilitate the physician a smooth transition from free space to contact and vice versa. For this purpose, the distance from the robot to the patient before contact was obtained using a 3D vision system mounted on the robot end-effector. However, note that such a smooth transition was not guaranteed with the aid of the aforementioned perceived stiffness, since the application was indeed a teleoperation that ultimately depended on the physician's commands.

4.1.2.4 Computer vision

As introduced in Chapter 2, the use of machine vision systems in robotic applications with an industrial perspective has been extensively carried out, for it gives the robot controller feedback about its environment, providing accuracy and flexibility.

On the one hand, some applications use eye-in-hand configurations, as in (Nilsen et al., 2019), where computer vision is used to improve the accuracy of a welding operation, or in (Muñoz-Benavent et al., 2019), where the tool of the robot is automatically changed thanks to an image-based visual servoing. On the other hand, other applications use eye-to-hand configurations, so as to localize pieces to perform operations such as pick and place (Astanin et al., 2017), or to keep track of the environment of the robot for safety purposes (Ferraro et al., 2012).

4.1.3 Proposal

The proposed approach presents several contributions to the previous literature, as discussed below.

4.1.3.1 Truly cooperative

In Chapter 2 it has already been established that, in contrast to the works (Villagrossi et al., 2017; Diao et al., 2018; Mohsin et al., 2017; Mohsin et al., 2019; Oba and Kakinuma, 2017; Tian et al., 2016; Sato et al., 2017; Segreto and Teti, 2019; Huang et al., 2020; Kharidege et al., 2017) mentioned above, the

proposed approach does not develop an automatic but rather a synergistic cooperation between the human operator and the robot system, combining their respective skills.

Specifically, this means the ability of the human operator to establish the regions of the workpiece surface that require the treatment, and the strength and precision of the robot system. In particular, in the application developed in this work the human operator guides the robot tool to the workpiece regions to be treated; while the robot system not only holds the tool and keeps the right tool orientation, but also guarantees a smooth approach to the workpiece and confines the tool within the allowed area close to the workpiece.

Furthermore, the degree of human-robot cooperation achieved by the proposed application is higher than that achieved by the works mentioned above in Section 4.1.2.3. In order to show this, the contributions referenced in Chapter 2 are discussed below in detail:

Both (Ochoa and Cortesão, 2019) and (Nemec et al., 2018) considered the interactions between a skilled operator and the robot only to program the task and, subsequently, the surface treatment was performed by the robot alone using an automatic mode of operation. Hence, in contrast to the proposed approach, the robot was not able to adapt to unforeseen changes of the target object, e.g., changes in position, shape, etc.

Both (Nemec et al., 2018) and (Gaz et al., 2018) used the robot system to hold the workpiece that was being polished by an external passive tool (Nemec et al., 2018) or by the human operator equipped with the tool (Gaz et al., 2018). Thus, the cooperation degree provided by the robot in these works is rather limited. In contrast, the proposed approach takes advantage of the robot system not only to hold the tool but also to keep, in real-time, the right tool orientation and to ensure a smooth approach to the workpiece surface. Moreover, the proposed approach is also suitable to treat large workpieces, e.g., the body of a car, which obviously would not be possible with the methods in (Nemec et al., 2018) and (Gaz et al., 2018).

In (Gracia et al., 2019) the perpendicularity between the robot tool and the workpiece surface was ensured by means of a force sensor mounted on the robot end-effector. However, this approach only applied to the contact phase. In contrast, the proposed approach ensures the aforementioned orthogonality for both contact and non-contact phases, i.e., the robot cooperation applies in both phases.

The medical application developed in (Santos and Cortesão, 2018) was es-

sentially a teleoperation task to perform an echography. Although the physician perceived the stiffness of the patient and surrounding areas through a haptic device, which was an aid to smooth the approach of the robot tool to the patient, such approach ultimately depended on the physician's commands. Thus, the cooperation degree provided by the robot in this work is rather limited. In contrast, the proposed approach uses the robot system in a more active and automatic manner, i.e., a smooth approach to the target object is guaranteed by means of a robot control system.

4.1.3.2 Camera network

A machine vision system is used in this work to get the location of the target workpiece in order to guarantee, whilst the user is guiding the tool to treat arbitrary regions of the workpiece surface, the perpendicularity between the workpiece surface and the tool and to ensure that the robot tool approaches the workpiece surface smoothly. As mentioned above, in contrast to other type of sensors such as force sensors, the vision system allows the robot to control not only the contact phase between the tool and the workpiece but also the non-contact phase. However, note that using computer vision implicitly assumes that the target workpice is rigid and known a priori.

The machine vision system developed in this research is based on a camera network, which has several advantages compared to the previous literature, as discussed below.

In (Villagrossi et al., 2017) and (Diao et al., 2018) a laser sensor and a vision system, respectively, mounted on the robot end-effector were used to previously scan and locate, also respectively, the target workpiece. However, these approaches had several drawbacks: a *previous phase* was required to scan or locate the workpiece; if the workpiece were substantially modified (e.g., if a larger workpiece were considered), the mentioned previous phase would have to be *reprogrammed*; the workpiece location was not updated in real-time and, hence, the robot system was not able to adapt to changes in the workpiece location, i.e., the workpiece had to be *static*; and the data obtained by the laser or camera was *local* to the robot end-effector and, hence, it was difficult to cover large areas.

In contrast, the proposed approach has the following advantages, some of which have already been discussed in Chapter 2: no previous phase is required; the method does not depend on the workpiece considered; the workpiece loca-

tion is updated in real-time and, hence, the workpiece can be moved; and the workpiece location is obtained globally, i.e., it is not constrained by the robot kinematics, and, hence, larger areas may be covered; and the camera network represents a *redundant* system that provides accuracy and robustness in case of occlusion.

As justified in Chapter 2, a network of three 3D sensors (Kinect cameras) with eye-to-hand configuration is used in this work, and the Iterative Closest Point (ICP) method is used for the registration process.

4.1.3.3 Smooth approach

In this work, a robust control system is presented to regulate the maximum velocity at which the tool of the robot system is allowed to approach the workpiece surface, being zero when the tool contacts the workpiece. Note that this control is not active either when the user moves the tool away from the workpiece or when the user moves it slowly toward the workpiece.

To the best of the authors' knowledge, (Santos and Cortesão, 2018) is the only work dealing with human-robot cooperation that developed a method to facilitate a smooth approach to the workpiece. In this application, the user teleoperated the robot by means of a haptic device while perceiving the *stiffness* of the target object and its surrounding areas, which were estimated using a force sensor and a 3D camera, respectively. However, this approach had several drawbacks compared to the proposed method: the smooth approach to the target object was *not ensured*, since perceiving the aforementioned stiffness did not guarantee that the user performed proper teleoperation commands; the perceived stiffness depended only on the *tool position*, i.e., the user did not get feedback about the tool speed, which is also very relevant to perform a smooth approach to the target object; and the perceived stiffness around the target object was estimated using a *local* sensor (i.e., a 3D camera mounted on the robot end-effector), which might suffer from occlusions and incomplete data.

In contrast, the proposed approach has the following advantages: the smooth approach to the target workpiece is guaranteed by an *approach control system*, which only uses degrees of freedom of the robot when it becomes active; not only the tool position but also the *tool speed* is considered to evaluate whether the robot approach control becomes active, i.e., the approach control can become active at any point of the workspace depending on the tool speed;

and the position and speed of the tool relative to the target workpiece are obtained from a redundant *global* sensor system (i.e., the camera network mentioned above) and, hence, the problems related to incomplete information are avoided.

4.1.3.4 Boundary constraints

Another distinctive feature of this work compared to the previous literature is that the proposed method confines the robot tool to an area close to the workpiece. Thus, when the operator guides the robot tool far away from the workpiece, a boundary constraint becomes active to prevent the tool from leaving the allowed area close to the workpiece. This approach, which represents a significant aid for the for the human operator when performing the surface treatment on the workpiece using the robot system, has two main advantages. On the one hand, unnecessary movements are avoided since the robot tool should not go far away from the workpiece while performing the surface treatment task. On the other hand, potential collisions between the robot tool and other objects within the robot workspace are prevented, i.e., the tool does not access forbidden areas of the robot environment where other objects are located.

4.1.3.5 Combination with automatic operation

Finally, another distinctive feature of this work compared to the previous literature is the combination of automatic operation together with manual operation, which adds more flexibility to the proposed method. Thus, the user is able to guide the tool of the robotic system to treat arbitrary regions of the workpiece (manual operation) and, when the user is not guiding the robot tool, a robot automatic operation is activated to perform the treatment in prior established regions.

4.2 Proposed method

4.2.1 General overview

Fig. 4.1 shows the general overview of the proposed method. In particular, three prioritized levels are considered to simultaneously accomplish several tasks. The first level, i.e., the highest priority level, is utilized both to limit

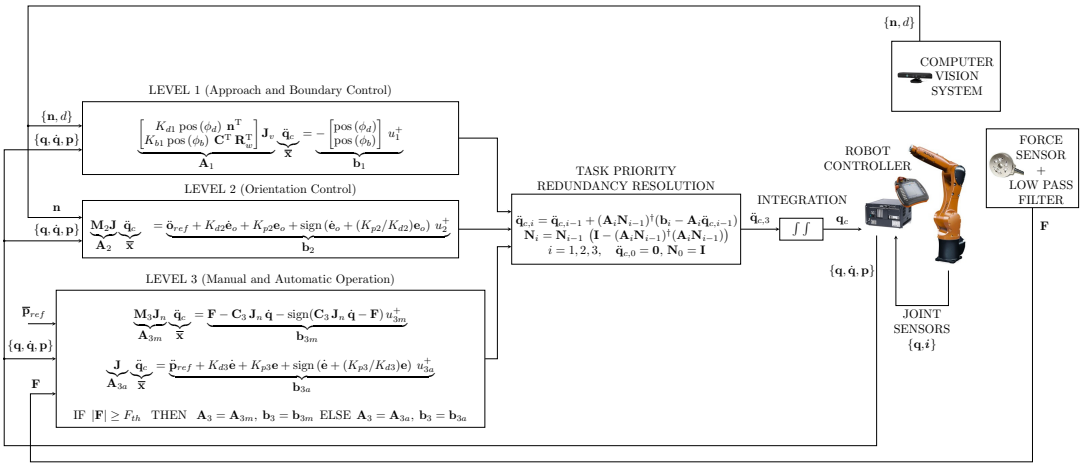


Figure 4.1: Block diagram of the proposed method.

the approach speed of the robot tool and to confine the tool to an area close to the workpiece. The second level is included to ensure that the tool is perpendicular to the workpiece surface. Lastly, the third level, i.e., the lowest priority level, is utilized to conduct the surface treatment on a particular part of the workpiece by means of a manual or an automatic mode of operation.

The following input information is considered for these levels: the robot pose $\mathbf{p}$ and the robot configuration and its derivative $\{\mathbf{q}, \dot{\mathbf{q}}\}$, which are obtained from the robot controller; the force vector $\mathbf{F}$ obtained from the *guidance sensor*, which is located at the robot tool; the data $\{d, \mathbf{n}\}$ obtained using machine vision, where d represents the length of the vector from the robot tool, see Fig. 4.2, to the nearest point of the workpiece, whereas $\mathbf{n}$ denotes the unit vector of the mentioned vector (note that $\mathbf{n}$ is normal to the workpiece surface as long as it is smooth at the nearest point to the tool); and the reference $\overline{\mathbf{p}}_{ref}$ for the tool position $\overline{\mathbf{p}} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}}$.

The equation $\mathbf{A}_i \overline{\mathbf{x}} = \mathbf{b}_i$ (3.4) for each priority level is obtained below, where $\overline{\mathbf{x}}$ corresponds to the commanded acceleration $\ddot{\mathbf{q}}_c$ for the robot system. The errors of these equations are minimized using (3.5) and (3.6), as shown in Fig. 4.1. Thus, the acceleration command $\ddot{\mathbf{q}}_{c,3}$ is double integrated to get the robot configuration command $\mathbf{q}_c$. Finally, the robot controller defines an inner control loop to track the commanded values taking into account the

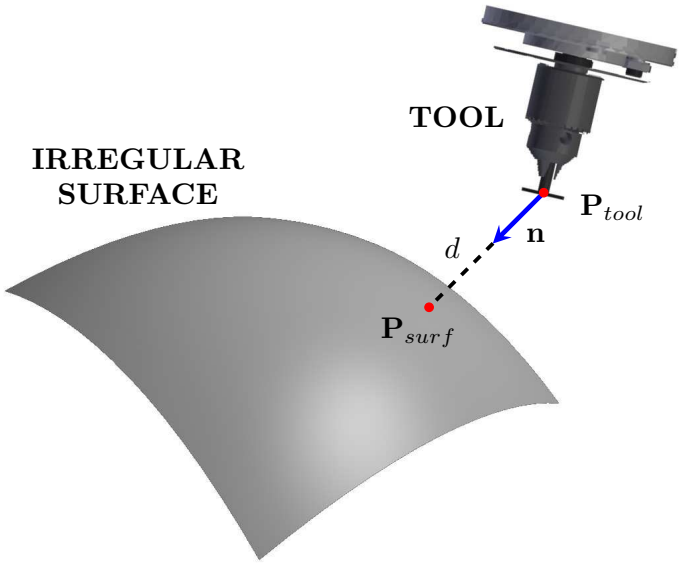


Figure 4.2: Graphical illustration of the information obtained from the machine vision system.

measurements of the joint angles $\mathbf{q}$ and joint currents $\boldsymbol{i}$. Hence, $\mathbf{d}_c$ stands for the inaccuracy of this inner loop, which is assumed to be bounded. However, note that the value of $\mathbf{d}_c$ does not need to be computed nor estimated since the proposed SMC-based controller, which is detailed below, is inherently robust against this error.

Keep in mind that in the contribution (García et al., 2020) the boundary constraint is not present, the first level is in charge of the control of the tool orientation, and the smooth approach is ensured by the second level of control.

4.2.2 Level 1: Approach and boundary control

In order to limit the approach speed of the robot tool to the workpiece, the following constraint is used:

$$\phi_d = \epsilon_d - d - K_{d1}\dot{d} \leq 0, \tag{4.1}$$

where ϵ_d represents a security margin between the tip of the tool and the workpiece surface and K_{d1} is a free design parameter that establishes the maximum approach speed allowed depending on the separation distance between the tool and the workpiece surface. Hence, the maximum allowed speed tends to zero as the mentioned separation distance tends to zero.

Considering that the motion of the workpiece (in case it is not static) is significantly slower than the motion of the robot system, the derivative of the distance d in (4.1) is readily obtained from the robot velocity as detailed below:

$$\begin{aligned}\dot{d} &= \left(\partial d/\partial \mathbf{q}\right)^{\mathrm{T}} \dot{\mathbf{q}} = \left(\left(\partial \overline{\mathbf{p}}/\partial \mathbf{q}\right)^{\mathrm{T}} \left(\partial d/\partial \overline{\mathbf{p}}\right)\right)^{\mathrm{T}} \dot{\mathbf{q}} \\ &= \left(\mathbf{J}_v^{\mathrm{T}}(-\mathbf{n})\right)^{\mathrm{T}} \dot{\mathbf{q}} = -\mathbf{n}^{\mathrm{T}}\mathbf{J}_v\dot{\mathbf{q}},\end{aligned} \tag{4.2}$$

where matrix $\mathbf{J}_v$ represents the top 3×3 submatrix of the Jacobian $\mathbf{J}$.

In order to confine the robot tool to an area close to the workpiece, the following constraint is considered:

$$\phi_b = \sigma_b + K_{b1}\dot{\sigma}_b \leq 0 \tag{4.3}$$

$$\sigma_b = -1 + \left(\left|\frac{x_w - x_c}{W}\right|^m + \left|\frac{y_w - y_c}{H}\right|^m + \left|\frac{z_w - z_c}{M}\right|^m\right), \tag{4.4}$$

where σ_b defines the boundary of the allowed area for the tool position as a superellipse, which looks like a rectangular prism with rounded corners,

m is a design parameter that establishes the rounding of the prism corners, parameters W, H and M define the length of each side of the prism, $\overline{\mathbf{p}}_w = \begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^{\mathrm{T}}$ is the tool position relative to the workpiece coordinate system, $\overline{\mathbf{p}}_c = \begin{bmatrix} x_c & y_c & z_c \end{bmatrix}^{\mathrm{T}}$ is the center of the superellipse[1] relative to the workpiece coordinate system and K_{b1} is a free design parameter in order to limit (similarly to the approach constraint) the approach speed of the tool to the boundary of the allowed area, i.e., the superellipse.

As before, considering that the motion of the workpiece is significantly slower than the motion of the robot system, the derivative of σ_b in (4.3) is obtained from the robot velocity as detailed below:

$$\begin{aligned} \dot{\sigma}_b &= \left(\partial\sigma_b/\partial\mathbf{q}\right)^{\mathrm{T}} \dot{\mathbf{q}} = \left(\left(\partial\overline{\mathbf{p}}_w/\partial\mathbf{q}\right)^{\mathrm{T}} \left(\partial\sigma_b/\partial\overline{\mathbf{p}}_w\right)\right)^{\mathrm{T}} \dot{\mathbf{q}} \\ &= \left(\left(\mathbf{R}_w^{-1}\mathbf{J}_v\right)^{\mathrm{T}} \mathbf{C}\right)^{\mathrm{T}} \dot{\mathbf{q}} = \mathbf{C}^{\mathrm{T}}\, \mathbf{R}_w^{\mathrm{T}}\, \mathbf{J}_v\, \dot{\mathbf{q}}, \end{aligned} \tag{4.5}$$

where $\mathbf{R}_w$ is the rotation matrix of the workpiece coordinate system with respect to the robot base coordinate system (note that the inverse of a rotation matrix is its transpose) and matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \begin{bmatrix} \dfrac{m\,\mathrm{sign}(x_w - x_c)\,|x_w - x_c|^{m-1}}{W^m} \\ \dfrac{m\,\mathrm{sign}(y_w - y_c)\,|y_w - y_c|^{m-1}}{H^m} \\ \dfrac{m\,\mathrm{sign}(z_w - z_c)\,|z_w - z_c|^{m-1}}{M^m} \end{bmatrix}. \tag{4.6}$$

In order to use the SMC detailed in Section 3.2 to satisfy the approach constraint in (4.1) and the boundary constraint in (4.3), the following second-order dynamical system (3.10) is considered:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{x} + \mathbf{d} + \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \mathbf{u}, \tag{4.7}$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $\mathbf{u} = \ddot{\mathbf{q}}_c$ and $\mathbf{d} = \mathbf{d}_c$.

[1] It has been assumed that the orientation of the superellipse matches the orientation of the workpiece coordinate system. However, if that would not be the case, the formula of the boundary constraint could be easily modified. Details omitted for brevity.

From Eqs. (3.15), (4.1) and (4.3), the control equation for Level 1 results in:

$$\begin{bmatrix} \text{pos}\,(\phi_d) & 0 \\ 0 & \text{pos}\,(\phi_b) \end{bmatrix} \mathbf{L_g}\phi_1\, \ddot{\mathbf{q}}_c = -\begin{bmatrix} \text{pos}\,(\phi_d) \\ \text{pos}\,(\phi_b) \end{bmatrix} u_1^+,$$
$$\rightarrow \mathbf{A}_1\ddot{\mathbf{q}}_c = \mathbf{b}_1, \tag{4.8}$$

where u_1^+ represents switching gain of the SMC, $\mathbf{b}_1$ and $\mathbf{A}_1$ denote the vector and matrix for the control equation of Level 1 and, according to (4.1)–(4.7), matrix $\mathbf{L_g}\phi_1$ is given by:

$$\mathbf{L_g}\phi_1 = \begin{bmatrix} (\partial\phi_d/\partial\mathbf{x})^{\mathrm{T}} \\ (\partial\phi_b/\partial\mathbf{x})^{\mathrm{T}} \end{bmatrix} \mathbf{g} = \begin{bmatrix} (\partial\phi_d/\partial\dot{\mathbf{q}})^{\mathrm{T}} \\ (\partial\phi_b/\partial\dot{\mathbf{q}})^{\mathrm{T}} \end{bmatrix}$$
$$= \begin{bmatrix} -K_{d1}\left(\partial\dot{d}/\partial\dot{\mathbf{q}}\right)^{\mathrm{T}} \\ K_{b1}\,(\partial\dot{\sigma}_b/\partial\dot{\mathbf{q}})^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} K_{d1}\,\mathbf{n}^{\mathrm{T}} \\ K_{b1}\,\mathbf{C}^{\mathrm{T}}\,\mathbf{R}_w^{\mathrm{T}} \end{bmatrix} \mathbf{J}_v. \tag{4.9}$$

4.2.3 Level 2: Orientation control

A key requirement for surface treatment operations is that the robot tool has to be orthogonal to the workpiece surface, that is, the Z-axis of the robot tool (see Fig. 4.2) must point in the direction of $\mathbf{n}$. Thus, the reference for the tool orientation is vector $\mathbf{n}$, which can be easily transformed (Siciliano and Khatib, 2008) to roll and pitch reference values, i.e., α_{ref} and β_{ref}. It is worth noting that there is no requirement for the yaw angle and, hence, it can be used, for instance, for tool guidance, see Section 4.2.4.1.

Thus, the control equation for Level 2 results in:

$$\mathbf{M_2}\mathbf{J}\ddot{\mathbf{q}}_c = \ddot{\mathbf{o}}_{ref} + K_{d2}\dot{\mathbf{e}}_o + K_{p2}\mathbf{e}_o + \text{sign}\,(\dot{\mathbf{e}}_o + (K_{p2}/K_{d2})\mathbf{e}_o)\, u_2^+$$
$$\rightarrow \mathbf{A}_2\ddot{\mathbf{q}}_c = \mathbf{b}_2, \tag{4.10}$$

where matrix $\mathbf{M}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ is used to affect only α and β angles (i.e., roll and pitch) of the pose vector $\mathbf{p}$; vector $\mathbf{o}_{ref} = \begin{bmatrix} \alpha_{ref} & \beta_{ref} \end{bmatrix}^{\mathrm{T}}$ represents the reference orientation; vector $\mathbf{e}_o = \mathbf{o}_{ref} - \begin{bmatrix} \alpha & \beta \end{bmatrix}^{\mathrm{T}}$ denotes the error of roll and pitch angles; K_{p2} and K_{d2} are the correction gains for the roll and pitch angles and their derivatives, respectively (note that $\dot{\alpha}$ and $\dot{\beta}$ can be

readily obtain from the robot equations in (3.2)); u_2^+ represents a switching gain; and $\mathbf{b}_2$ and $\mathbf{A}_2$ denote the vector and matrix for the control equation of Level 2.

It is worth noting that (4.10) represents a hybrid controller, where the last switching term is used to cancel out the last term in (3.3) and, hence, the computation of the Jacobian derivative is avoided. Note that this hybrid controller in sort uses conventional SMC, whose proof of convergence can be found in (Utkin et al., 2009).

4.2.4 Level 3: Modes of operation

This level is included to conduct the surface treatment on a particular part of the workpiece by means of a manual or an automatic mode of operation. Specifically, if the guidance sensor detects a significant force value ($|\mathbf{F}| \geq F_{th}$, where F_{th} represents a threshold) the manual operation becomes active, i.e., the user guides the robot tool using the guidance sensor. Otherwise, the automatic operation is activated, and the robot tool follows a reference value. The control equations for both cases are detailed below.

Note that this level controls the tool yaw angle and the tool position, either by the manual or the automatic mode of operation, whereas the tool pitch and roll angles are established in Level 2.

4.2.4.1 Manual operation

In a similar way to other works that use human-robot interaction (Yao et al., 2018; Li et al., 2018), the human operator guides the tool exerting forces that are transformed by the following admittance controller into the desired values of the tool speed:

$$\mathbf{M}_3 \, \dot{\mathbf{v}}_n + \mathbf{C}_3 \, \mathbf{v}_n = \mathbf{F}, \tag{4.11}$$

where vector $\mathbf{v}_n = \mathbf{J}_n \dot{\mathbf{q}}$ denotes the tool speed with respect to the tool coordinate system, matrix $\mathbf{J}_n$ represents the *geometric* Jacobian (Siciliano and Khatib, 2008) with respect to the tool coordinate system, vector $\mathbf{F}$ contains the force measurements of the guidance sensor with respect to the tool coordinate system and the controller gains $\mathbf{C}_3$ and $\mathbf{M}_3$ are 6×6 diagonal matrices representing the virtual damping and inertia, respectively.

Taking into account that $\mathbf{v}_n = \mathbf{J}_n\dot{\mathbf{q}}$, Eq. (4.11) results in:

$$\mathbf{M}_3\,\mathbf{J}_n\,\ddot{\mathbf{q}} = \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{M}_3\,\dot{\mathbf{J}}_n\,\dot{\mathbf{q}}. \tag{4.12}$$

Similarly to Level 2, the computation of the Jacobian derivative in (4.12) can be avoided using the following hybrid controller:

$$\begin{aligned}\mathbf{M}_3\,\mathbf{J}_n\,\ddot{\mathbf{q}}_c = \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \text{sign}(\mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{F})\,u^+_{3m}\\ \rightarrow \mathbf{A}_{3m}\ddot{\mathbf{q}}_c = \mathbf{b}_{3m},\end{aligned} \tag{4.13}$$

where u^+_{3m} represents a switching gain and $\mathbf{b}_{3m}$ and $\mathbf{A}_{3m}$ denote the vector and matrix for the control equation of Level 3 when the manual operation is active.

4.2.4.2 Automatic operation

In a similar way to (4.10), the following controller is used to follow the reference pose $\mathbf{p}_{ref}$ when the automatic operation is active:

$$\begin{aligned}\mathbf{J}\ddot{\mathbf{q}}_c =& \ddot{\mathbf{p}}_{ref} + K_{d3}\dot{\mathbf{e}} + K_{p3}\mathbf{e} + \text{sign}\,(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e})\,u^+_{3a}\\ &\rightarrow \mathbf{A}_{3a}\ddot{\mathbf{q}}_c = \mathbf{b}_{3a}\end{aligned} \tag{4.14}$$

where $\mathbf{e} = \mathbf{p}_{ref} - \mathbf{p}$ denotes the tool pose error; K_{p3} is the correction gain for the pose error; K_{d3} is the correction gain for the pose error derivative; u^+_{3a} represents a switching gain; and $\mathbf{b}_{3a}$ and $\mathbf{A}_{3a}$ denote the vector and matrix for the control equation of Level 3 when the automatic operation is active.

During the automatic operation the tool yaw angle is kept still, i.e., its reference value γ_{ref} corresponds to the angle value at the moment the automatic operation was activated. Moreover, the reference value $\overline{\mathbf{p}}_{ref}$ for the tool position is established according to the following four stages that are cyclically repeated to treat each point on the workpiece surface:

1. The robot tool is placed at a certain distance from the point on the workpiece surface using a fast approach trajectory.

2. The tool is taken to the point on the workpiece surface using a slow linear trajectory.

3. The tool is kept still during a time lapse to ensure that the surface treatment is properly completed.

4. The robot tool is moved away from the workpiece surface using a slow linear trajectory.

Since the above cycle can be interrupted by the manual operation, when the robot system goes back to automatic operation, it resumes the stage where it was before the interruption.

In the next chapter, an approach based on sliding-mode is proposed for reference tracking in robot visual servoing.

4.3 Control algorithm

4.3.1 Code of the control algorithm

Table 4.1 shows the pseudo-code of the algorithm developed in this work.

Each line of the code is explained as follows. The first line of code, i.e., LC1, is used to update the readings from the sensors[2]. LC2 is used to compute the orientation angles of $\mathbf{n}$, which is provided by the computer vision. LC3 and LC4 are used to compute the robot pose and its derivative from the robot kinematics. LC5 to LC7 are used to compute the constraint functions of the inequalities in Level 1. LC8, LC9, LC10 and LC11 are used to compute the time derivative of several signals. LC12 to LC15 are used to compute the orientation and pose errors and their derivatives (the automatic operation provides the reference pose $\mathbf{p}_{ref}$ using the procedure outlined in Section 4.2.4.2). LC16 and LC17 are used to compute the matrix and vector, respectively, for the control equation of Level 1. LC18 and LC19 are used to compute the matrix and vector, respectively, for the control equation of Level 2. LC20 to LC26 are used to compute the matrix and vector for the control equation of Level 3. LC27 to LC31 are used to compute the solution of the commanded accelerations that minimizes the errors of the control equations of the three levels. LC32 and LC33 are used to compute the commanded positions by integrating twice the commanded accelerations. LC34 is used to send the commanded values to the robot controller. LC35 to LC38 are used to update the previous value of several signals in order to use them in the next iteration.

[2]It is assumed that the electronics of the F/T sensor (guidance sensor) has filtered the force measurements and that all disturbance forces have already been compensated, e.g., the weight and inertia of the robot tool.

Table 4.1: Code of the algorithm

Algorithm executed at sampling time of T_s seconds	
$[\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}, \mathbf{n}, d]$ =GetRobotStateAndForcesAndVisionData();	
$\mathbf{o}_{ref}$ =OrientationOfVector(**n**);	
$\mathbf{p} = \mathbf{l}(\mathbf{q})$;	// Eq. (3.1)
$\dot{\mathbf{p}} = \mathbf{J}\dot{\mathbf{q}}$;	// Eq. (3.2)
$\phi_d = \epsilon_d - d + K_{d1}\mathbf{n}^{\mathrm{T}}\mathbf{J}_v\dot{\mathbf{q}}$;	// Eqs. (4.1),(4.2)
$\sigma_b = -1 + \left(\left\|\frac{x_w - x_c}{W}\right\|^m + \left\|\frac{y_w - y_c}{H}\right\|^m + \left\|\frac{z_w - z_c}{M}\right\|^m\right)$;	// Eq. (4.4)
$\phi_b = \sigma_b + K_{b1}\mathbf{C}^{\mathrm{T}}\,\mathbf{R}_w^{\mathrm{T}}\,\mathbf{J}_v\,\dot{\mathbf{q}}$;	// Eqs. (4.3),(4.5)
$\dot{\mathbf{o}}_{ref} = (\mathbf{o}_{ref} - \mathbf{o}_{ref,prev})/T_s$;	// Derivative
$\ddot{\mathbf{o}}_{ref} = (\dot{\mathbf{o}}_{ref} - \dot{\mathbf{o}}_{ref,prev})/T_s$;	// Derivative
$\dot{\mathbf{p}}_{ref} = (\mathbf{p}_{ref} - \mathbf{p}_{ref,prev})/T_s$;	// Derivative
$\ddot{\mathbf{p}}_{ref} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}}_{ref,prev})/T_s$;	// Derivative
$\mathbf{e}_o = (\mathbf{o}_{ref} - \mathbf{M}_2\mathbf{p})$;	// Orientation error
$\dot{\mathbf{e}}_o = (\dot{\mathbf{o}}_{ref} - \mathbf{M}_2\dot{\mathbf{p}})$;	// Derivative of orientation error
$\mathbf{e} = (\mathbf{p}_{ref} - \mathbf{p})$;	// Pose error
$\dot{\mathbf{e}} = (\dot{\mathbf{p}}_{ref} - \dot{\mathbf{p}})$;	// Derivative of pose error
$\mathbf{A}_1 = \begin{bmatrix} K_{d1}\mathrm{pos}\,(\phi_d)\ \mathbf{n}^{\mathrm{T}} \\ K_{b1}\mathrm{pos}\,(\phi_b)\ \mathbf{C}^{\mathrm{T}}\,\mathbf{R}_w^{\mathrm{T}} \end{bmatrix} \mathbf{J}_v$;	// Eqs. (4.8),(4.9)
$\mathbf{b}_1 = -\begin{bmatrix} \mathrm{pos}\,(\phi_d) \\ \mathrm{pos}\,(\phi_b) \end{bmatrix}\ u_1^+$;	// Eq. (4.8)
$\mathbf{A}_2 = \mathbf{M}_2\mathbf{J}$;	// Eq. (4.10)
$\mathbf{b}_2 = \ddot{\mathbf{o}}_{ref} + K_{d2}\dot{\mathbf{e}}_o + K_{p2}\mathbf{e}_o + \mathrm{sign}\,(\dot{\mathbf{e}}_o + (K_{p2}/K_{d2})\mathbf{e}_o)\,u_2^+$;	// Eq. (4.10)
if $\mathbf{F} \geq F_{th}$ **then**	
$\quad \mathbf{A}_3 = \mathbf{M}_3\mathbf{J}_n$;	// Eq. (4.13)
$\quad \mathbf{b}_3 = \mathbf{F} - \mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathrm{sign}(\mathbf{C}_3\,\mathbf{J}_n\,\dot{\mathbf{q}} - \mathbf{F})\,u_{3m}^+$;	// Eq. (4.13)
else	
$\quad \mathbf{A}_3 = \mathbf{J}$;	// Eq. (4.14)
$\quad \mathbf{b}_3 = \ddot{\mathbf{p}}_{ref} + K_{d3}\dot{\mathbf{e}} + K_{p3}\mathbf{e} + \mathrm{sign}\,(\dot{\mathbf{e}} + (K_{p3}/K_{d3})\mathbf{e})\,u_{3a}^+$;	// Eq. (4.14)
end	
$\ddot{\mathbf{q}}_{c,1} = \mathbf{A}_1^{\dagger} b_1$;	// Eq. (3.5), $i = 1$
$\mathbf{N}_1 = \mathbf{I} - \mathbf{A}_1^{\dagger}\mathbf{A}_1$;	// Eq. (3.6), $i = 1$
$\ddot{\mathbf{q}}_{c,2} = \ddot{\mathbf{q}}_{c,1} + (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{b}_2 - \mathbf{A}_2\ddot{\mathbf{q}}_{c,1})$;	// Eq. (3.5), $i = 2$
$\mathbf{N}_2 = \mathbf{N}_1(\mathbf{I} - (\mathbf{A}_2\mathbf{N}_1)^{\dagger}(\mathbf{A}_2\mathbf{N}_1))$;	// Eq. (3.6), $i = 2$
$\ddot{\mathbf{q}}_{c,3} = \ddot{\mathbf{q}}_{c,2} + (\mathbf{A}_3\mathbf{N}_2)^{\dagger}(\mathbf{b}_3 - \mathbf{A}_3\ddot{\mathbf{q}}_{c,2})$;	// Eq. (3.5), $i = 3$
$\mathbf{N}_3 = \mathbf{N}_2(\mathbf{I} - (\mathbf{A}_3\mathbf{N}_2)^{\dagger}(\mathbf{A}_3\mathbf{N}_2))$;	// Eq. (3.6), $i = 3$
$\dot{\mathbf{q}}_c = \ddot{\mathbf{q}}_{c,3}\,T_s + \dot{\mathbf{q}}_{c,prev}$;	// Integration
$\mathbf{q}_c = \dot{\mathbf{q}}_c\,T_s + \mathbf{q}_{c,prev}$;	// Integration
SendToJointControllers($\mathbf{q}_c$);	
$\mathbf{o}_{ref,prev} = \mathbf{o}_{ref}$;	// For next iteration
$\dot{\mathbf{o}}_{ref,prev} = \dot{\mathbf{o}}_{ref}$;	// For next iteration
$\mathbf{p}_{ref,prev} = \mathbf{p}_{ref}$;	// For next iteration
$\dot{\mathbf{p}}_{ref,prev} = \dot{\mathbf{p}}_{ref}$;	// For next iteration

Note that the kinematic function and jacobian matrices of the robot arm can be readily obtained from its Denavit-Hartenberg parameters as detailed in (Corke, 2017).

The computation of one iteration of the algorithm in Table 4.1 (compiled C code) takes around 0.015 milliseconds for the case in Section 4.4.

4.3.2 Design of the control algorithm parameters

The practical guidelines to choose the parameters of the control algorithm are summarized in the following steps:

1^{st}) The sampling period T_s is chosen as small as possible but ensuring: that the noise introduced by the numerical derivatives in Table 4.1 is negligible; and that the SMC frequency $f_{SMC} = (2Ts)^{-1}$ is lower than the bandwidth of the low-level joint controllers (otherwise, the SMC actions would not be properly "followed" by the robot system).

2^{nd}a) The bandwidth of the kinematic control performed in Level 1 (given by K_{d1} and K_{b1}), Level 2 (given by K_{p2} and K_{d2}) and Level 3 (given by $\mathbf{M}_3$ and $\mathbf{C}_3$ or K_{p3} and K_{d3}) should be significantly lower than the SMC frequency f_{SMC} for stability reasons.

2^{nd}b) For Level 1, the parameters K_{d1} and K_{b1} are chosen small enough to reduce the chattering effect of the SMC, but satisfying the condition indicated in step 2^{nd}a).

2^{nd}c) For Level 2, the parameters K_{p2} and K_{d2} are chosen large enough to obtain a fast and damped enough response, respectively, but satisfying the condition indicated in step 2^{nd}a).

2^{nd}d) For the automatic operation in Level 3, the parameters K_{p3} and K_{d3} are chosen large enough to obtain a fast and damped enough response, respectively, but satisfying the condition indicated in step 2^{nd}a).

2^{nd}e) For the manual operation in Level 3, firstly the diagonal elements of matrix $\mathbf{C}_3$ are chosen small enough to obtain the desired sensitivity of the tool guidance and, subsequently, the diagonal elements of matrix $\mathbf{M}_3$ are chosen small enough to obtain a fast enough response. Moreover, $\mathbf{M}_3$ and $\mathbf{C}_3$ must satisfy the condition indicated in step 2^{nd}a).

3^{rd}) For all three levels, the switching gains $\{u_1^+, u_2^+, u_{3m}, u_{3a}^+\}$ are empirically tuned to be as small as possible to alleviate the chattering effect of the SMC, but ensuring that the sliding mode behavior remains effective.

4^{th}a) For Level 1, the parameters $\{m, W, H, M\}$ of the superellipse are chosen to fit the allowed area for the specific robot application at hand, whereas the parameter ϵ_d of the approach constrain is chosen large enough to cater for possible inaccuracies in the robot control.

4^{th}b) For the manual operation in Level 3, the parameter F_{th} should be small enough to properly activate the tool guidance when the user is guiding the robot tool, but not too small, since that could cause undesired activations due to force sensor noise.

4^{th}c) For the automatic operation in Level 3, the time lapse used to ensure that the surface treatment is properly completed has to be established depending on the requirements of the actual surface treatment task, where the knowledge of an experienced operator could be very useful.

4.4 Experiments

4.4.1 Experimental platform

The experimental platform used in this work is shown in Fig. 4.3, which was composed of: a 6R robot arm (Kuka KR6 Agilus), whose Denavit-Hartenberg parameters are shown in Table 4.2; a tool consisting of a spot repair sander (Mirka AROS-B 150NV) placed at the robot end-effector using a self-developed adapter; an Axia80 F/T (Force/Torque) sensor used as guidance sensor, which is attached between the end-effector of the robot arm and the sander; a cylinder of 29x29x23 mm used as sanding disc; 3 RGB-D cameras (Microsoft Kinect); and a workpiece consisting of a car door.

An external computer was used to implement the algorithm detailed in Section 4.3.1. Moreover, the robot arm, F/T sensor and external computer communicated by means of an Ethernet switch. In addition, the RGB-D cameras were connected to serial ports of the external computer.

The maximum workpiece position error given by the used 3D camera network was around 1mm, which was acceptable for the proposed application.

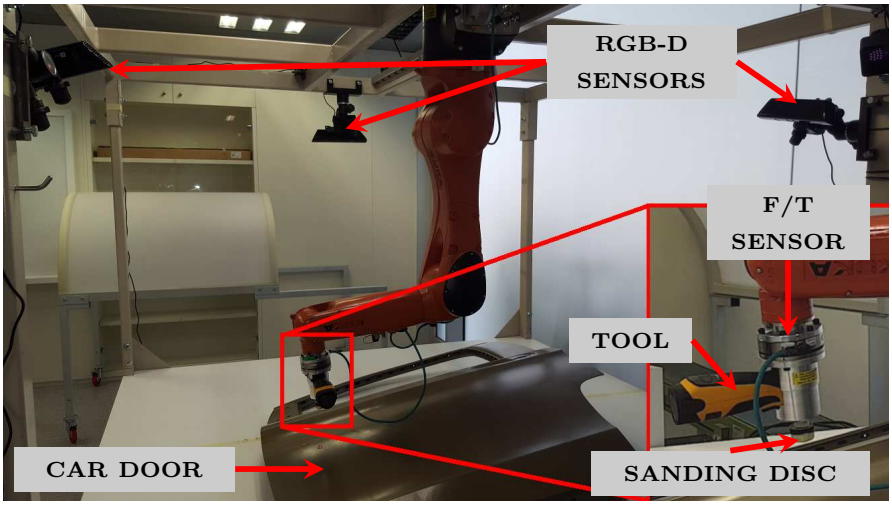


Figure 4.3: Experimental platform used for the real experimentation: a 6R robot arm, a F/T sensor, 3 RGB-D cameras , an industrial sander and a car door.

Table 4.2: Denavit-Hartenberg parameters for the robot used in the experiments ($d_{tool} = 0.08$)

Link i	θ_i (rad)	d_i (m)	a_i (m)	α_i (rad)
1	q_1	-0.4	0.025	$\pi/2$
2	q_2	0	-0.455	0
3	q_3	0	-0.035	$-\pi/2$
4	q_4	-0.42	0	$\pi/2$
5	q_5	0	0	$-\pi/2$
6	q_6	$-0.08 - d_{tool}$	0	π

However, more accurate depth sensors could be used for applications requiring more precision.

4.4.2 Values of the parameters

The values used for the control algorithm parameters are given below. They were established according to the practical guidelines given in Section 4.3.2.

i) Sampling period: $T_s = 0.01$ s.

ii) Parameters of Level 1 (Section 4.2.2): $\epsilon_d = 2$ mm, $K_{d1} = 2.5$, $m = 4$, $W = 0.2$, $H = 0.25$, $M = 0.35$, $K_{b1} = 1.6$, and $u_1^+ = 0.65$.

iii) Parameters of Level 2 (Section 4.2.3): $K_{p2} = 1.5$, $K_{d2} = 1.8$ and $u_2^+ = 0.01$.

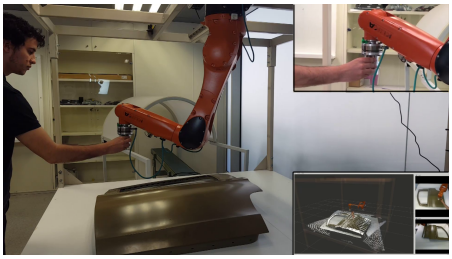
iv) Parameters of Level 3 (Section 4.2.4): $\mathbf{M}_3 = 10\,\mathbf{I}$, $\mathbf{C}_3 = 70\,\mathbf{I}$, $F_{th} = 1$, $u_{3m}^+ = 0.01$, $u_{3a}^+ = 0.01$, $K_{p3} = 2$ and $K_{d3} = 4.2$.

4.4.3 Results

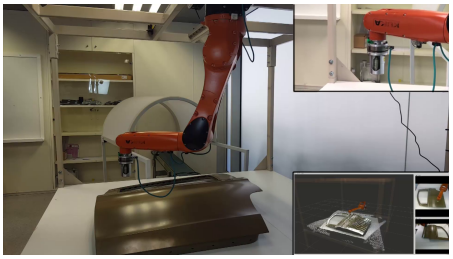
Three experiments were conducted to study the behavior of: the approach constraint, the boundary constraint and the combination of manual and automatic modes of operation, respectively. Hence, for the first two experiments only the manual operation was considered.

Therefore, in order to analyze the behavior of the robot approaching the surface of the workpiece (i.e., the car door) a first experiment was conducted, see the `video` (Video: Chapter 4, Experiment 1, 2020) (in order to obtain a clearer view, the sander was detached from the robotic arm end-effector). Several frames of this video recording are shown in Fig. 4.4: at around 12s, see Fig. 4.4(a), the human operator places a weight of about 0.175 Kg in the robotic arm end-effector and, thus, the tool guidance of the manual operation becomes active; in the interval 19s–26s, see Fig. 4.4(b) and Fig. 4.4(c), the end-effector of the robotic arm goes down towards the workpiece while its vertical speed is progressively reduced; and at around 32s, see Fig. 4.4(d), the robot arm stops its movement, keeping the security distance with the workpiece surface.

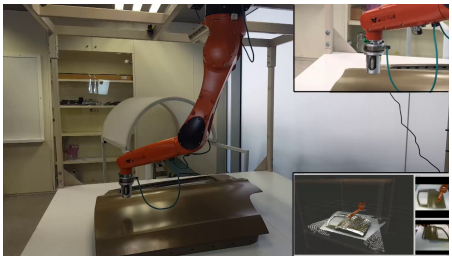
Next, several graphs are presented to show the quantitative performance of the first experiment. In particular, the distance of separation between the



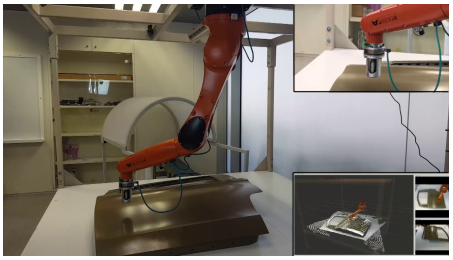
(a) 0m12s (time instant 2s in the graph)



(b) 0m19s (time instant 9s in the graph)



(c) 0m26s (time instant 16s in the graph)



(d) 0m32s (time instant 22s in the graph)

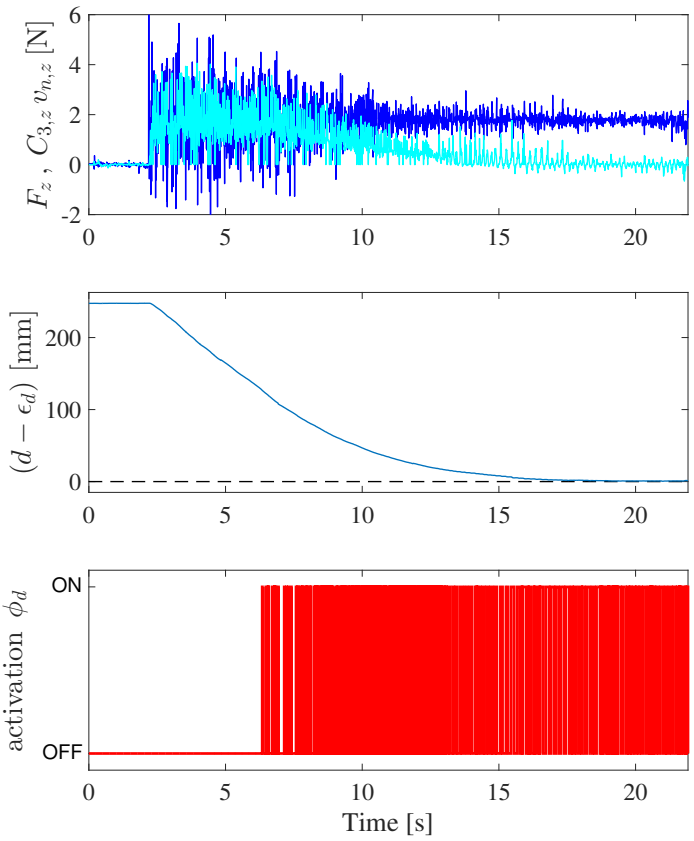
Figure 4.4: Frames of the video of the first experiment.

end-effector of the robotic arm and the workpiece is initially around 250mm, see the middle graph in Fig. 4.5(a). Then, at around 2.3s the operator places the small weight in the end-effector of the robotic arm, see the negative vertical force F_z detected by the guidance sensor in the top graph of Fig. 4.5(a), and the aforementioned separation is progressively reduced. Note that the reduction rate is roughly constant during the interval 2.3s–6.3s, see the middle graph of Fig. 4.5(a). This is because the tool guidance of the manual operation in Level 3 converts the small weight, which is constant, to a downward speed $v_{n,z}$ for the end-effector of the robotic arm, as shown in the top graph of Fig. 4.5(a). Subsequently, as shown in the bottom graph of Fig. 4.5(a), at about 6.3s the approach constraint becomes active. From then, the distance reduction rate and the negative vertical velocity $v_{n,z}$ of the end-effector of the robotic arm are progressively reduced due to the approach constraint in Level 1. Finally, at about 22s the robot arm approximately reaches the security distance, i.e., $d = \epsilon_d$, and stops its movement. A graph relating the distance and the distance reduction rate is shown in Fig. 4.5(b), where it can be appreciated that the system has a switching behavior (as usual in SMC) around the boundary of the approach constraint, which is obtained from (3.25) as $\dot{d} = -(1/K_2)(d-\epsilon_d)$.

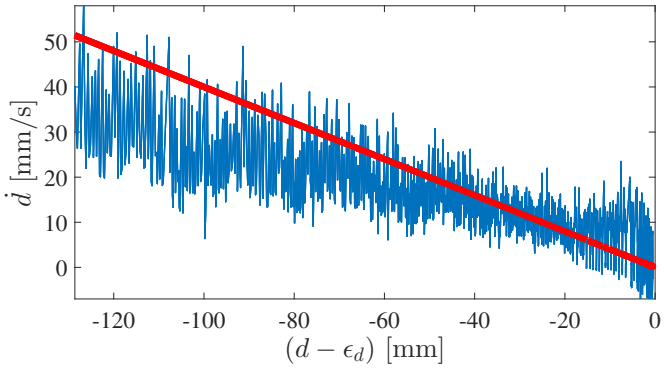
The tool orientation angles for the first experiment are shown in Fig. 4.6. In particular, the first and second graphs show that the values of roll and yaw angles are very similar to the reference values supplied by the machine vision system, which is due to the orientation control implemented in Level 2. Moreover, the yaw angle is approximately constant, see the bottom graph. This is due to the fact that the small weight placed in the end-effector of the robotic arm produces no torque in the Z-axis of the end-effector of the robotic arm and, hence, no guidance is performed for the yaw angle in Level 3.

The Cartesian position of the end-effector of the robotic arm in the first experiment is represented in Fig. 4.7, where it can be noted that the trajectory followed by the robot end-effector is approximately a vertical straight line of about 0.25m.

The control signals in the first experiment are presented in Fig. 4.8, where the commanded accelerations computed by each control level are shown. In particular, the contribution of Level 2 is very little since only small corrections are required to keep the end-effector of the robotic arm perpendicular to the workpiece surface as it moves downwards. Moreover, the contribution of Level 1 starts when the approach constraint becomes active, which occurs at around 6.3s. Furthermore, the contribution of Level 3 starts at about 2.3s,



(a) Top graph: dark-blue line, negative vertical force F_z detected by the guidance sensor; and light-cyan line, negative vertical velocity $v_{n,z}$ of the robot end-effector multiplied by the virtual damping coefficient $C_{3,z}$ (note that the unit of the velocity is m/s, whereas the unit of the velocity multiplied by the damping coefficient is Newton, see Section 4.2.4.1 and Eq. (4.11)). Middle graph: distance of separation between the end-effector of the robotic arm and the workpiece surface. Bottom graph: activation of the approach constraint.



(b) Distance reduction rate of the robot end-effector towards the workpiece as a function of distance (thin-blue line) and boundary given by the approach constraint (thick-red line).

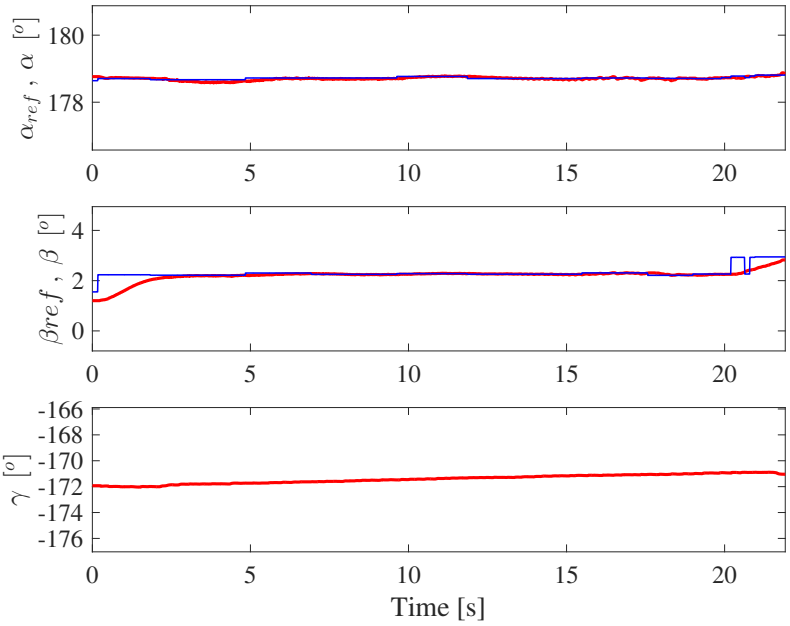


Figure 4.6: Tool orientation angles in the first experiment. From top to bottom: roll, pitch and yaw angles. In the first two graphs: thin line, reference values supplied by the machine vision system; thick line, actual angle values.

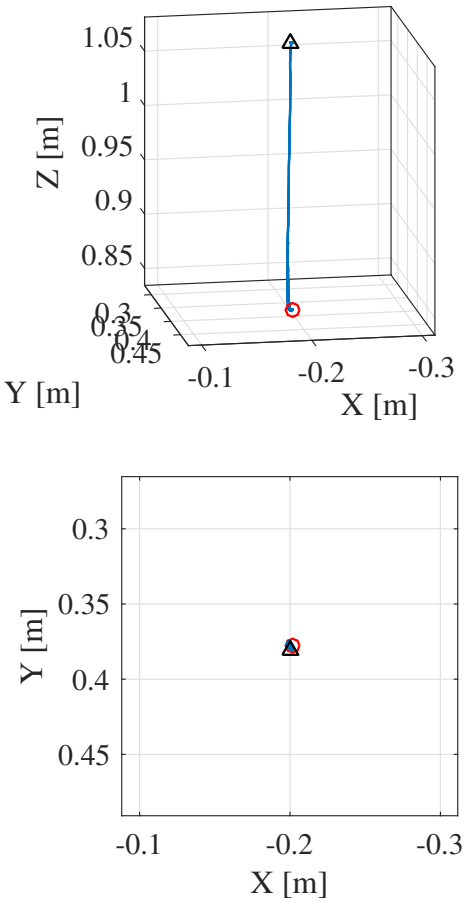


Figure 4.7: Trajectory followed by the robot end-effector in the first experiment (triangle and circle symbols denote the initial and final positions, respectively).

i.e., when the human operator places the small weight in the end-effector of the robotic arm.

A second experiment was performed to analyze the behavior of the boundary constraint, see the video (Video: Chapter 4, Experiment 2, 2020). Several frames of this video recording are shown in Fig. 4.9: at around 16s, see Fig. 4.9(a), the human operator is guiding the robotic arm tool close to the workpiece; at around 26s, see Fig. 4.9(b), the human operator moves up the robot tool and the boundary constraint becomes active, preventing the tool from leaving the allowed area, i.e., the rectangular prism with rounded corners relative to the workpiece; at around 1m09s, see Fig. 4.9(c), the workpiece (i.e., the car door) is relocated; and at around 1m33s, see Fig. 4.9(d), the human operator is again guiding the tool away from the workpiece and the boundary constraint becomes active, preventing the tool from leaving the allowed area, which has been properly updated after the workpiece relocation due to the real-time information supplied by the 3D camera network.

Next, several graphs are presented to show the quantitative performance of the second experiment. Fig. 4.10 shows the functions and activation of the boundary constraint. Note that the constraint is activated during six intervals (see the bottom graph) but the tool position $\overline{\mathbf{p}}_w$ (relative to the workpiece) is properly confined within the allowed area since the value of σ_b is always less than zero, see the top graph and (4.4).

The tool orientation angles for the second experiment are shown in Fig. 4.11. As before, the first and second graphs show that the values of roll and yaw angles are very similar to the reference values supplied by the machine vision system, which is due to the orientation control implemented in Level 2. Moreover, the yaw angle is mainly modified at the beginning of the experiment, which is due to the tool guidance of the manual operation. In this sense, Fig. 4.12 shows the behavior of the tool guidance in Level 3, where it can be seen that the forces exerted by the human operator are properly followed by the tool velocities except at some intervals (see the second and third graphs) due to the activation of the boundary constraint, as shown in Fig. 4.10. That is, the tool does not follow the operator forces if that means leaving the allowed area.

The tool position in the second experiment is represented in Fig. 4.13. Note that, despite the tool guidance performed by the user, the trajectory followed by the robot tool is properly confined within the allowed area, whose boundary is given by a superellipse, which looks like a rectangular prism with

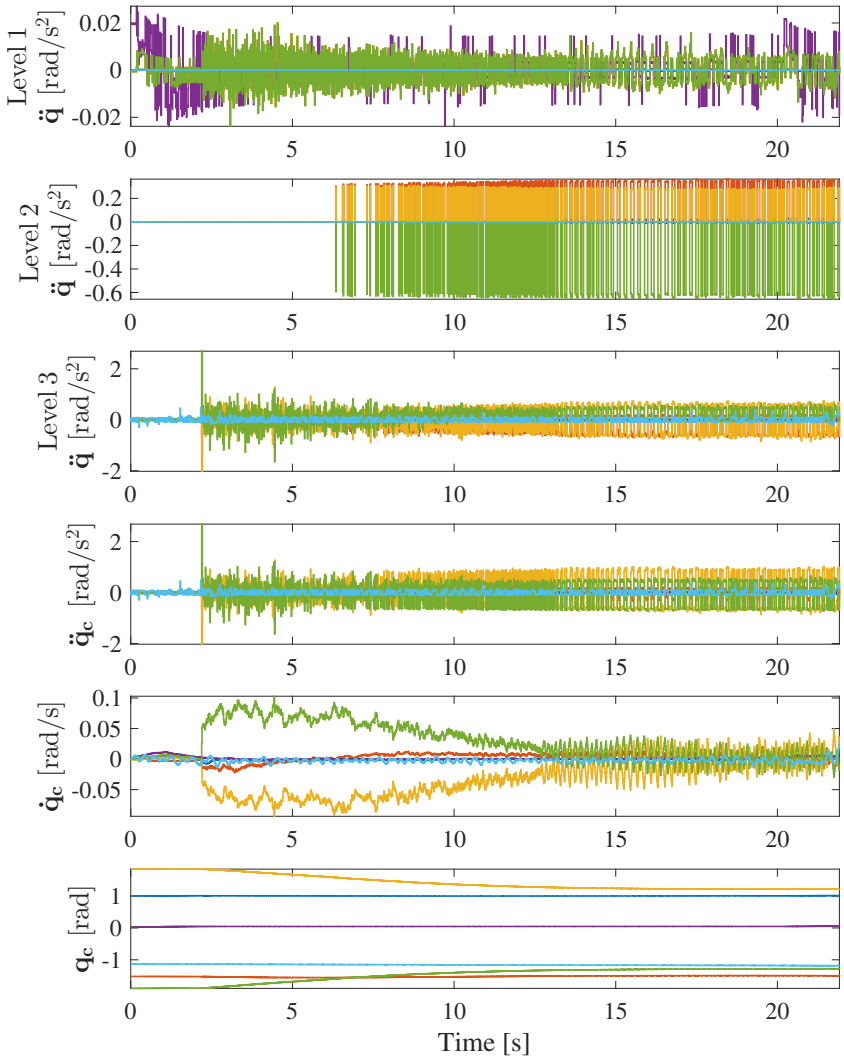
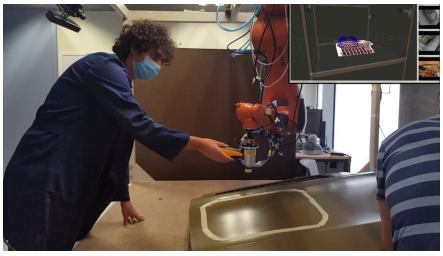


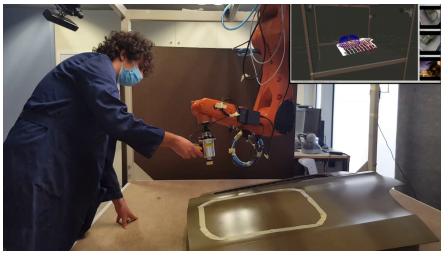
Figure 4.8: Control signals in the first experiment. From top to bottom: commanded accelerations computed by each control level; joint accelerations, velocities and positions to be sent to the robot controller. In the graphs, a different color is used for each robot joint, i.e., from the first to the sixth joint: blue, brown, yellow, magenta, green and cyan.

(a) 0m16s (time instant 10s in the graph)

(b) 0m26s (time instant 20s in the graph)



(c) 1m09s (time instant 63s in the graph)



(d) 1m33s (time instant 87s in the graph)

Figure 4.9: Frames of the video of the second experiment.

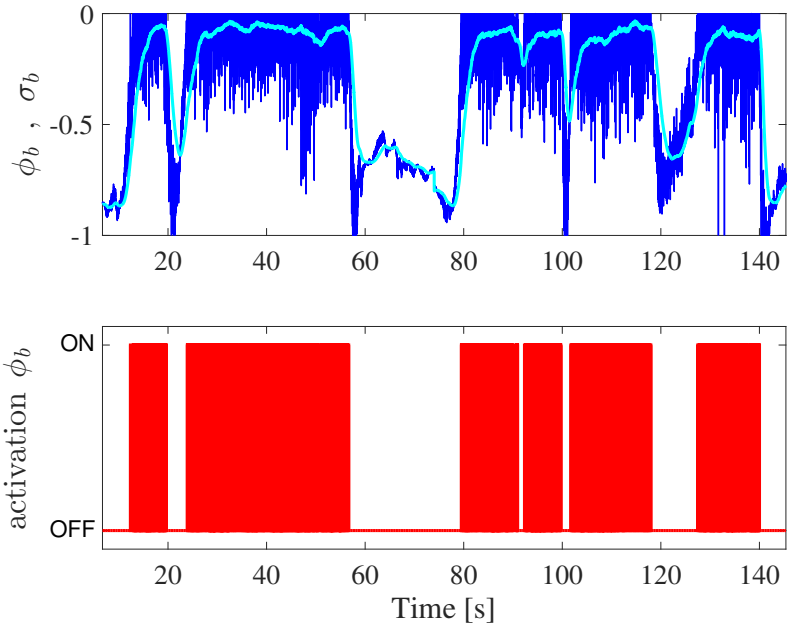


Figure 4.10: Boundary constraint in the second experiment: top graph, constraint functions ϕ_b (dark-blue) and σ_b (light-cyan); bottom graph, activation of the boundary constraint.

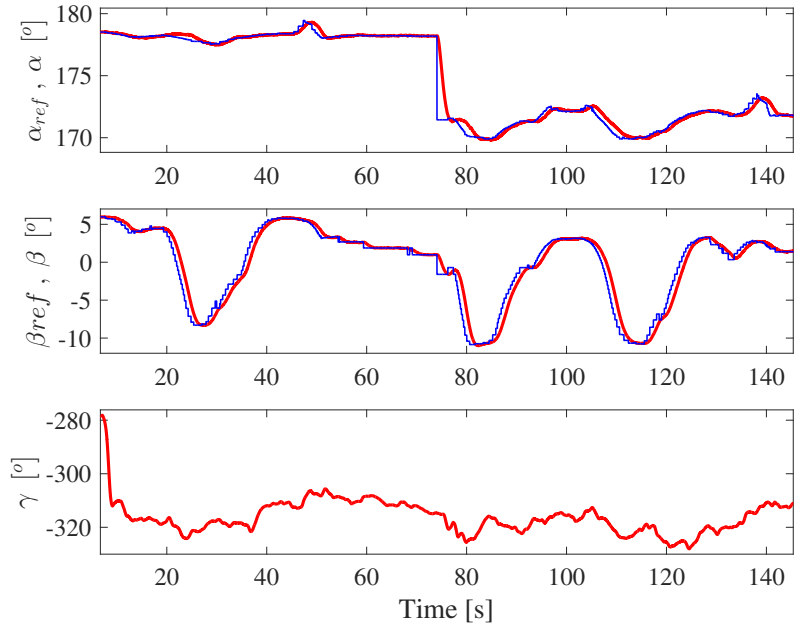


Figure 4.11: Tool orientation angles in the second experiment: α, β and γ. In the first two graphs: thin line, reference values supplied by the machine vision system; thick line, actual angle values.

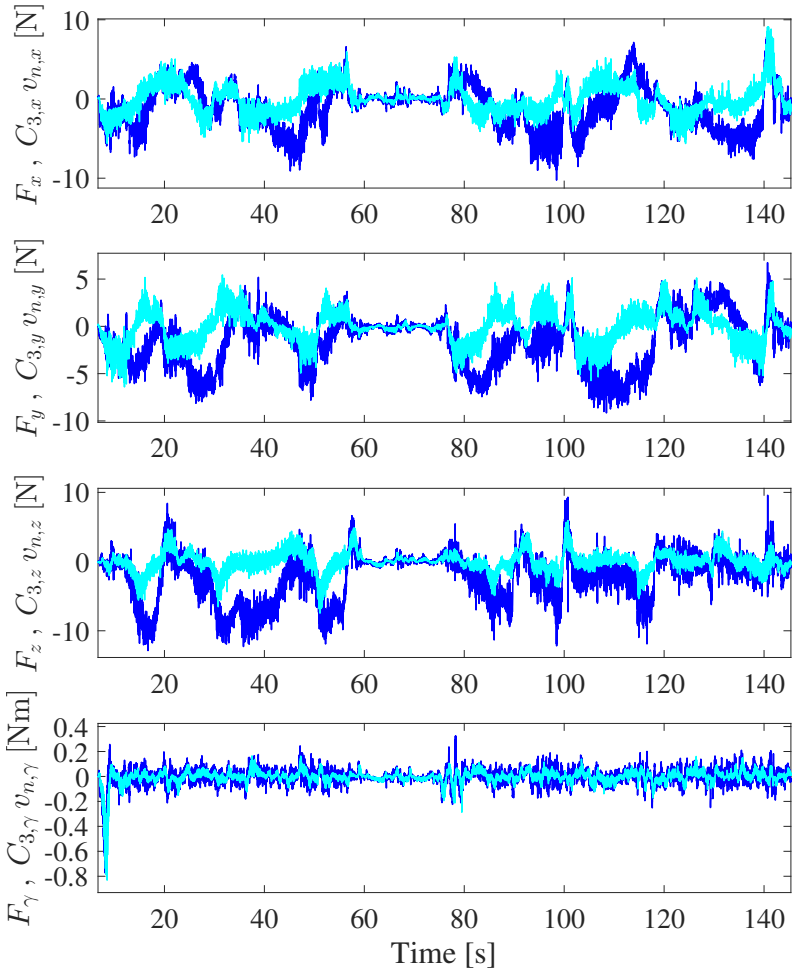


Figure 4.12: Tool guidance in the second experiment: tool velocities (multiplied by $\mathbf{C}_d$) in light-cyan and forces of the human operator in dark-blue. From top to bottom: linear X, linear Y, linear Z and angular Z components of the vectors (all four components are relative to the tool coordinate system).

rounded corners as shown in the figure.

The control signals in the second experiment are presented in Fig. 4.14, where the commanded accelerations computed by each control level are shown. Note that the commanded acceleration computed by Level 1 (see the top graph) is non-zero when the boundary constraint is active (see the bottom graph in Fig. 4.10). Moreover, the commanded acceleration computed by Level 2 (see the second graph) is also non-zero when the boundary constraint is active, which is due to the fact that the orientation control has to compensate the corrections introduced by Level 1 in order to keep the right tool orientation. Finally, note that the commanded acceleration computed by Level 3 (see the third graph) is approximately zero in the interval 60s-75s, which corresponds to the part of the experiment where the user is not guiding the tool, see Fig. 4.12.

A third experiment was performed to show the utility of combining the two modes of operation, i.e., manual and automatic operation, see the `video` (Video: Chapter 4, Experiment 3, 2020). Several frames of this video recording are shown in Fig. 4.15: at around 43s, see Fig. 4.15(a), the robotic arm is using the automatic operation to perform the sanding on a pre-established point of the workpiece surface (two pre-established points of the workpiece surface were cyclically used by the automatic operation); in the interval 1m03s–1m12s, see Fig. 4.15(b) and Fig. 4.15(c), the human operator uses the tool guidance to treat other regions of the workpiece surface; at around 2m22s, see Fig. 4.15(d), the robotic arm is again using the automatic operation to perform the sanding on a pre-established point of the workpiece surface; at around 3m00s, see Fig. 4.15(e), the workpiece (i.e., the car door) is relocated; and at around 3m25s, see Fig. 4.15(f), the human operator is again guiding the tool to perform the sanding on an arbitrary region of the workpiece surface, which is done properly despite the previous workpiece relocation due to the real-time information supplied by the 3D camera network.

It is worth remarking that the proposed approach is able to dynamically adapt to arbitrary changes of the workpiece location due to the real-time information supplied by the 3D camera network. This feature has been shown both in the second and third experiments. In particular, as explained above, in the second experiment the workpiece (i.e., the car door) is arbitrarily relocated at around 1m09s and, subsequently, the boundary constraint is properly activated to prevent the tool from leaving the allowed area, which is properly updated after the workpiece relocation. Furthermore, as mentioned above, in

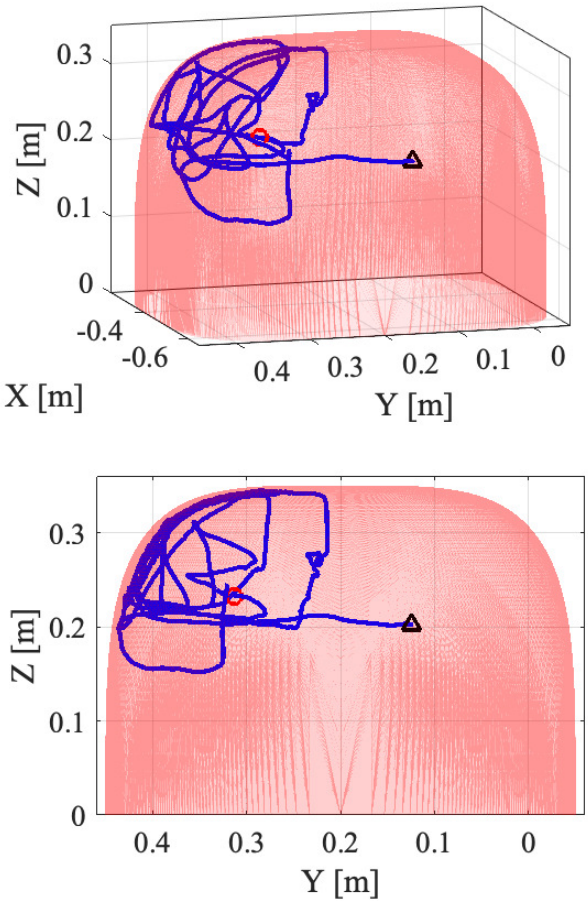


Figure 4.13: Trajectory of the tool position in the second experiment (triangle and circle symbols denote the initial and final positions, respectively) and mesh representing the boundary of the allowed area.

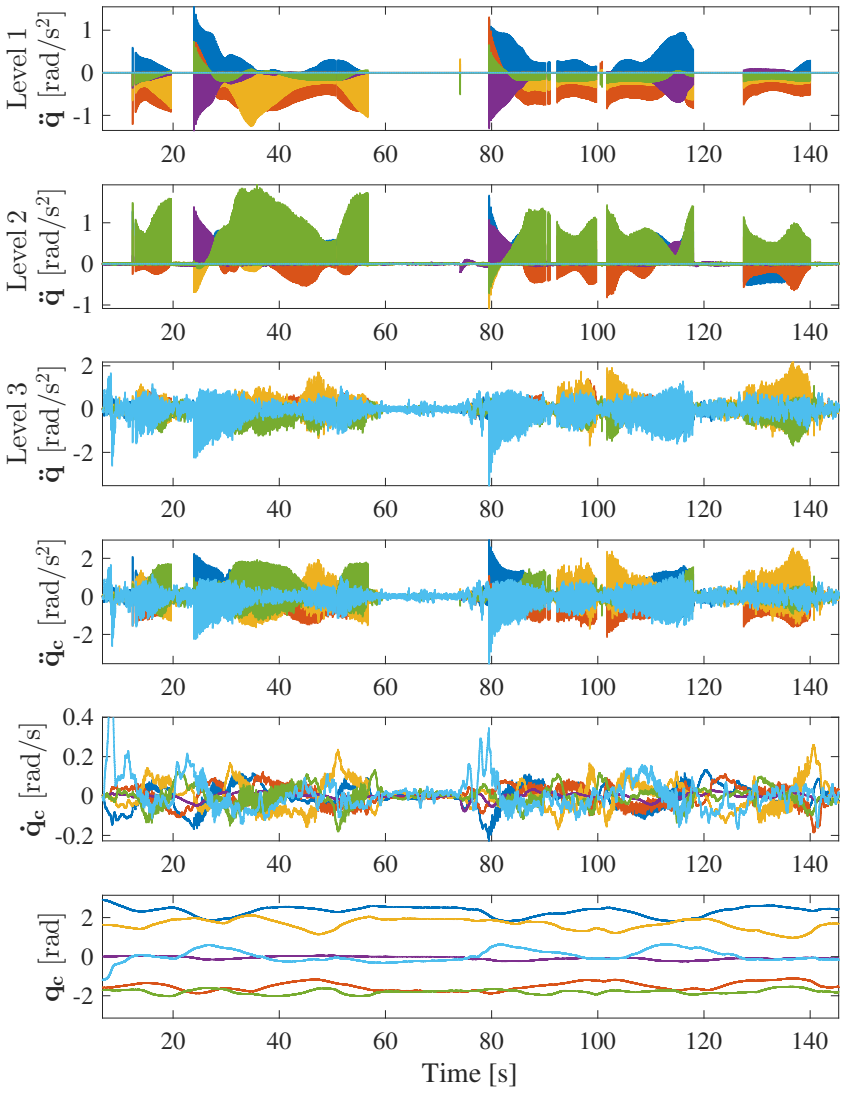
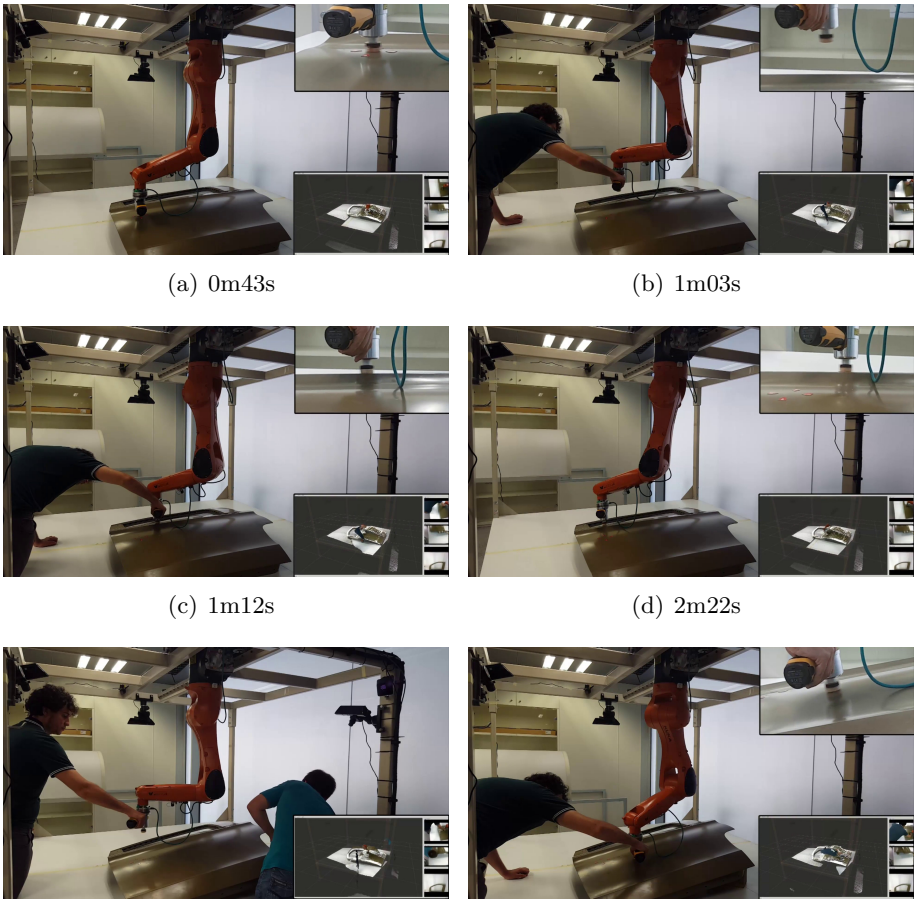


Figure 4.14: Control signals in the second experiment. From top to bottom: commanded accelerations computed by each control level; joint accelerations, velocities and positions to be sent to the robot controller.



(a) 0m43s (b) 1m03s

(c) 1m12s (d) 2m22s

(e) 3m00s (f) 3m25s

Figure 4.15: Frames of the third experiment recording.

the third experiment the workpiece is arbitrarily relocated at around 3m00s and, subsequently, the human operator guides the tool to perform the sanding on a specific region of the workpiece surface, which is done properly. That is, the tool approaching and perpendicularity are properly controlled by the robot system despite the previous workpiece relocation.

Chapter 5

Bimanual robot control using assisted teleoperation for surface treatment tasks

5.1 Introduction

5.1.1 Objective

As it has been introduced in Chapters 1 and 2, Human-Robot Interaction is a rich field of research due to the impossibility of automating certain operations completely using current technology, and also because of the advantages offered by the involvement of human abilities even in operations with a high degree of automation.

Teleoperation permits different degrees of human implication in the application, while keeping a remote control architecture which is interesting for work environments, as it eliminates unnecessary human presence, thus increasing safety and also efficiency in regards to space requirements.

And regarding robotic systems, bimanual solutions, i.e., the collaboration of two robotic manipulators working on the same workpiece to accomplish a shared goal, are of deep interest, due to their higher versatility in comparison with single robot arm systems and the possibility of emulating human-like manual operations.

Thus, this chapter presents a control architecture for a bimanual robotic

system in order to conduct a surface treatment task in which the human user partially teleoperates both robot arms. Note that the target task is complex and mainly non-automated (Kieselbach et al., 2019).

5.1.2 State of the Art

5.1.2.1 Assisted teleoperation in robotics

Since the foundations of teleoperation and the general trends of shared-control architectures have already been described in Chapter 2, a more detailed study of the literature is presented below:

As explained in Chapter 2, the two extremes of the shared-control spectre are direct control (i.e. the robot directly executes the human's commands) and supervisory control (automated operations complemented with high-level human decisions). Some examples can be found of these two extremes, for instance (Liu et al., 2019), a case of direct control where the contribution is focused on the processing of neural signals to teleoperate a dual arm system; (Bandala et al., 2019), where both cases are present: first, the direct control of the mobile platform of the robot and then, the supervisory control of a dual-arm manipulator system, where the human operator selects points for the robotic system to grasp and cut automatically, and indicates the moment when the automated task should start; or (Isop et al., 2019), where the aerial exploration of indoor environments is carried out using a supervisory control approach, so the teleoperator chooses whether the aerial robot should inspect an object, explore an area or go to the next area, whereas the control algorithm manages every subtask making the task possible.

A more assisted version of direct control is telepresence, already introduced in Chapter 2, and consisting of developing interfaces which make this direct control easier for the human user. Telepresence is a strong trend in recent research developments, with the introduction of virtual or augmented reality (Gorjup et al., 2019; Solanes et al., 2020), visual interfaces (Yoon et al., 2018) and haptic devices (Selvaggio et al., 2019) or the combination of these different elements (Girbés-Juan et al., 2021; Clark et al., 2019; Saracino et al., 2020) to direct control teleoperation. A particular case can be found in (Nicolis et al., 2018), where one arm of a bimanual robot is teleoperated to grasp a target object, while the other develops an automatic task of visual-servoing to keep the object in sight of a camera and avoiding occlusions, thus making the teleoperation easier.

Since telepresence relies heavily on the user's skills, a whole body of research focuses on imposing restrictions to the position references the human operator can command, by incorporating virtual barriers, such as Virtual Fixtures (Chen et al., 2020b) in surgery, an example of which can be found in (Kapoor et al., 2005), where the user's position references are automatically modified to fit an allowed area, and Haptic Guidance applications, where a haptic system avoids the user from commanding reference positions beyond certain limits, as can be seen in (Selvaggio et al., 2019; Abi-Farraj et al., 2020).

Finally, shared-control architectures where the robot plays an active role and the human still keeps the initiative, as it is the case in this work, fully exploit the potential of Human-Robot Interaction, specially when they are combined with reactive control methods, which can provide relatively low computational cost and real-time control. Some representative contributions in this trend are discussed below:

In (Selvaggio et al., 2018), a system composed of two robot manipulators sharing the same workspace is developed, where one of the robot manipulators carries out an automatic visual task, while the other is teleoperated with a shared-control approach, in order to grasp a target object. The teleoperation is assisted by the control algorithm in the two defined situations: when the gripper is far from the target object, the control algorithm controls the gripper orientation in order to avoid the system constraints (joint limits, singularities and collisions with the other robot) and leaves the human user in control of the gripper translation, although this is also limited by a conventional haptic guidance which helps the user to avoid the constraints; whereas when the gripper is close to the target object, the control algorithm manages all the necessary Degrees of Freedom (DoF) to make sure that the gripper always points towards the object, and so the teleoperator can use the remaining DoF to move the gripper in a sphere centered around the target object. The present work shares part of the approach of (Selvaggio et al., 2018), with an assisted teleoperation where the control algorithm manages different DoF depending on the evolution of certain parameters so as to ensure that necessary restrictions are met. However, in (Selvaggio et al., 2018) part of the shared-control relies on conventional haptic guidance, whereas the present work ensures the accomplishment of all the restrictions with a multitask control architecture with different priority levels.

In (Suarez et al., 2020), an aerial manipulation application is developed and tested. In this case, the human user is expected to carry out the tele-

operation of two compliant manipulators attached to the aerial robot system, while the control application manages the variables of the flight using nested PID controllers and also applies a compliant joint and force control to the manipulators.

Although the present work shares with (Suarez et al., 2020) combining a reactive control approach with the dexterity and flexibility provided by a human teleoperator, there is a significant difference, for the present work assists the teleoperation by directly limiting the DoF available for the teleoperation of the bimanual robotic system and using those DoF to ensure certain constraints, while (Suarez et al., 2020) assists the teleoperation mainly by making sure that the conditions necessary for the teleoperation are met, but the human user exerts an almost completely direct teleoperation over the manipulators.

Finally, in (Brantner and Khatib, 2021) an assisted subaquatic teleoperation of a humanoid exploration robot with a bimanual manipulator system is developed. Similarly to the present work, a multitask control architecture with different priority levels is implemented, so a lower priority control law can use just the DoF that higher priority control laws are not using. In (Brantner and Khatib, 2021), this architecture is developed in order to make sure that certain restrictions are met (in this case, joint limits, self collision and obstacle avoidance) while the human operator teleoperates the robot arms to perform the manipulation task, and, in the lowest priority level, the robot body and arms postures are automatically controlled in order to optimize the position of the body with respect to the hands and to optimize the inertia of the robot hands by adjusting the position of the arms. However, (Brantner and Khatib, 2021) mainly relies on conventional continuos control laws, whereas the present work utilizes advanced Sliding Mode Control (SMC) techniques.

5.1.2.2 Bimanual robotics

As introduced in Chapter 2, there is a wide variety of contributions involving dual-arm robotic systems, as they present several advantages in comparison to one-arm robot manipulators, including their relative resemblance to human behaviour, which makes them easier to relate to and to command.

Some of these contributions are goal-coordinated dual-arm robotic applications (same task and no physical interaction): for instance, in (Selvaggio et al., 2018; Nicolis et al., 2018) similar manipulation applications are presented, where one robotic arm is teleoperated to manipulate a workpiece while

the other performs a fully automated visual servoing algorithm to keep track of the workpiece while avoiding collisions and other constraints, with no physical interaction between the two robot manipulators; and in (Sepúlveda et al., 2020), where two robots pick different aubergines simultaneously, and, when it is necessary, one arm helps the other by removing obstacles for the picking task.

Differently to these contributions, bimanual robotics applications, such as the one developed in this work, imply that not only the robotic manipulators work to achieve the same goal, but also that they resort to phyisical interaction to accomplish this goal. Thus, bimanual robotics find a wide range of applications, such as: manipulation of deformable objects (Chatzilygeroudis et al., 2020; Garcia-Camacho et al., 2020; Sintov et al., 2020), objects with unknown shape (Mitash et al., 2020; Clark et al., 2019) or objects the geometry of which requires two grasping points (Salehian et al., 2018; Suarez et al., 2020; Wu et al., 2019), emulation of human bimanual tasks (Rakita et al., 2019; García et al., 2019; Qu et al., 2019), assistive robotics (Joshi et al., 2020; Li et al., 2019), assembly operations (Chen et al., 2020a; Zimmermann et al., 2020; Parigi Polverini et al., 2019), surgery tasks (Zhong et al., 2019) and simultaneous manipulation and cutting (Bandala et al., 2019), manipulation and fastening (Makris et al., 2017) or manipulation and surface treatment (Girbés-Juan et al., 2021), which is the case of this work.

A significant number of bimanual robotics contributions develop fully automated applications, presenting diverse approaches, such as offline training and machine learning (Joshi et al., 2020; García et al., 2019), solutions based on motion planning (Mitash et al., 2020; Sintov et al., 2020; McConachie et al., 2020; Zimmermann et al., 2020; Liang et al., 2020), sometimes including specific situations where a reactive control method (for instance, force control or impedance control) is activated in order to improve the motion planner performance (Chen et al., 2020a; Parigi Polverini et al., 2019; Zhong et al., 2019) and applications which rely mainly on closed-loop control methods (Wu et al., 2019; Li et al., 2019).

As it has already been explained in Chapter 2, Human-Robot Interaction is specially interesting in bimanual robotics applications. Of course, there exist contributions which include some degree of HRI and bimanual robotic configurations where the human does not interact with the bimanual task in itself, but rather with the robotic system as a whole. This is the case of (Makris et al., 2017), where the HRI consists of slowing down or stopping the

operation of the bimanual robotic system in order to allow the presence of the human in the same workspace; or the supervisory control approaches involving bimanual tasks cited in Section 5.1.2.1.

However, the framework of the present work is that of Human-Robot Interaction involving a bimanual operation.

In this same framework, (Ibarguren et al., 2020) proposes an operation typically performed by two human operators, the transportation of a large workpiece, where one of the operators is substituted by a bimanual robotic system mounted on a mobile platform. To carry out the operation, the workpiece can be moved along a pre-determined trajectory, and the human operator retains the initiative to start and stop the movement as well as to decide arbitrary adaptations of this trajectory, thanks to a conventional impedance control. Furthermore, certain limits for the arbitrary deformations of the trajectory are pre-determined too.

Thus, in a similar way to the present contribution, in (Ibarguren et al., 2020) the Human Robot Interaction benefits from the relatability and flexibility of a bimanual robotic configuration, and also the human operator retains the initiative for the main aspects of the operation, while being guided to perform it correctly, thanks to a reactive control. However, differently to (Ibarguren et al., 2020), in this contribution the human user (the teleoperator), can command totally arbitrary movements to both robot arms, the constraints are respected thanks to an advanced control architecture and there is a higher degree of coordination between the robot arms, since they are in close contact during the surface treatment operation.

On the other hand, a more similar related work can be found in (Brantner and Khatib, 2021). Although this contribution has already been addressed above in Section 5.1.2.1, there are some similarities to the present work regarding the bimanual task in itself which are worth recalling. Besides the existence of several layers of control with different levels of priority, the bimanual task is also subject of shared-control: a position control keeps the robot hands orientation constant and a force control keeps contact with the manipulated surface, thus leaving the teleoperator with 2 DoF to command, i.e., movement on a plane. This approach is similar to the one taken in the present contribution for one of the robot manipulators, as it will be explained in Section5.2.

5.1.3 Proposed approach

This work develops a method to perform surface treatment tasks using a bimanual robotic system, i.e., two robot arms cooperatively performing the task. In particular, one robot arm, namely Workpiece Robot (WR), holds the workpiece while the other robot arm, namely Surface Treatment Robot (STR), has the treatment tool attached to its end-effector.

In this way, some robot coordinates are teleoperated by the human user, while the remaining robot coordinates are automatically controlled. In particular, the user teleoperates all the six coordinates of the WR in order to put the workpiece in a proper position and orientation for the task. Moreover, the teleoperator commands two linear coordinates of the STR to move the treatment tool on the workpiece surface in order to apply the surface treatment. Furthermore, a Force/Torque (F/T) sensor attached to the STR end-effector is used to automatically adapt the STR tool in order to attain the desired pressure between the tool and the workpiece as well as to keep the tool orientation orthogonal to the workpiece surface.

In addition, to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace, e.g., to avoid collisions with other objects in the environment.

As mentioned above, the theory used in this work to develop the bimanual robot control relies on SMC as well as task prioritization.

5.2 Proposed approach

The goal of this work is to develop a robot control using the SMC in Section 3.2 so that two robot manipulators can be simultaneously teleoperated by a human user to conduct a surface treatment task. One of the robots, namely *workpiece robot* (WR), holds the workpiece while the other robot, namely *surface treatment robot* (STR), has the surface treatment tool and a F/T sensor attached to its end-effector, so the human operator can adapt the position and orientation of the workpiece and, at the same time, command the movement of the surface treatment tool over the workpiece surface.

In order to achieve this, some coordinates of both robot arms are controlled automatically:

– WR: The WR control keeps the workpiece center inside the allowed workspace whose boundary is given by a superellipsoid, which resembles a rectangu-

lar cuboid with smoothed corners. Moreover, for safety, the WR control limits the angular positions that the workpiece can achieve while the user teleoperates the WR. Additionally, in case that the WR is redundant (i.e., the WR has more than six DoF), the WR control uses the redundant DoF to "push" the WR to a home configuration for increased safety.

– STR: The STR control keeps the robot tool center inside the allowed area on the workpiece, whose boundary is given by a modified superellipse, which resembles a rectangle with smoothed corners. Moreover, the STR control maintains the desired pressure and orthogonality to the workpiece needed to perform the treatment task, using the data from an F/T sensor, coined as treatment sensor. The remaining DoF of the STR are available for the human user to teleoperate the STR on the surface of the workpiece.

In the equations below, subscripts "w" and "s" are used to denote WR and STR, respectively.

5.2.1 System tasks

A different control architecture is developed for each robot arm of the bimanual application proposed in this work.

On the one hand, the following four prioritized tasks are considered for the WR control:

W1) The first level (high-priority task) includes the *inequality constraints* that must be satisfied to keep the workpiece center within the allowed region, whose boundary is given by a superellipsoid as mentioned above.

W2) The second level (medium-high-priority task) includes the *inequality constraints* required for the workpiece to keep its angular position within previously specified limits.

W3) The third level (medium-low-priority task), a hybrid SMC, ensures a reference tracking so the human user can teleoperate the WR.

W4) The fourth level (low-priority task), which only applies for the case of *redundant* robots, is used to keep the robot close to a *home* configuration.

The input to these tasks is the WR state $\{\mathbf{q}_w, \dot{\mathbf{q}}_w\}$ (and $\mathbf{p}_w$, which is obtained from the WR kinematic function) and the reference $\mathbf{p}_{w,ref}$ generated by the human teleoperator for the WR pose vector, whereas each task gives an acceleration equality whose square error must be minimized.

On the other hand, three tasks with the same approach of priority levels are considered for the STR control:

S1) The first level (high-priority task) includes the *inequality constraints* that must be satisfied to keep the surface treatment tool center in the allowed area on the workpiece surface, whose boundary is given by a modified superellipse as mentioned above.

S2) The second level (medium priority task) includes the *equality constraints* that must be satisfied at all times to properly perform the treatment on the workpiece with the STR tool. In particular, equality constraints are defined to exert the desired pressure between the STR tool and the workpiece being treated, and to keep the tool orientation orthogonal to the workpiece surface.

S3) The third level (low priority task), a hybrid SMC, ensures a reference tracking so the human user can teleoperate the STR.

The input to these tasks is the STR state $\{\mathbf{q}_s, \dot{\mathbf{q}}_s\}$ (and $\mathbf{p}_s$, which is obtained from the STR kinematic function), the force vector $\mathbf{F}$ with the measurements of the F/T sensor and the reference $\overline{\mathbf{p}}_{sb,ref}$ generated by the human teleoperator for the 2D position of the STR tool on the workpiece, whereas each task gives an acceleration equality whose square error must be minimized.

In particular, the acceleration equality for the first and second levels of the WR control and the first level of the STR control are obtained below using the one-side SMC presented in Section 3.2.2, whereas the acceleration equality for the second level of the STR control are obtained below using the conventional SMC presented in Section 3.2.1.

Fig. 5.1 depicts the overview block diagram of the proposed control for the WR and STR.

It is worth noting that the collaboration between the human user and the bimanual robotic system in the surface treatment task consists in the *teleoperation* of both robot arms by the user. That is, the teleoperator commands all the six coordinates of the WR in order to put the workpiece in a proper

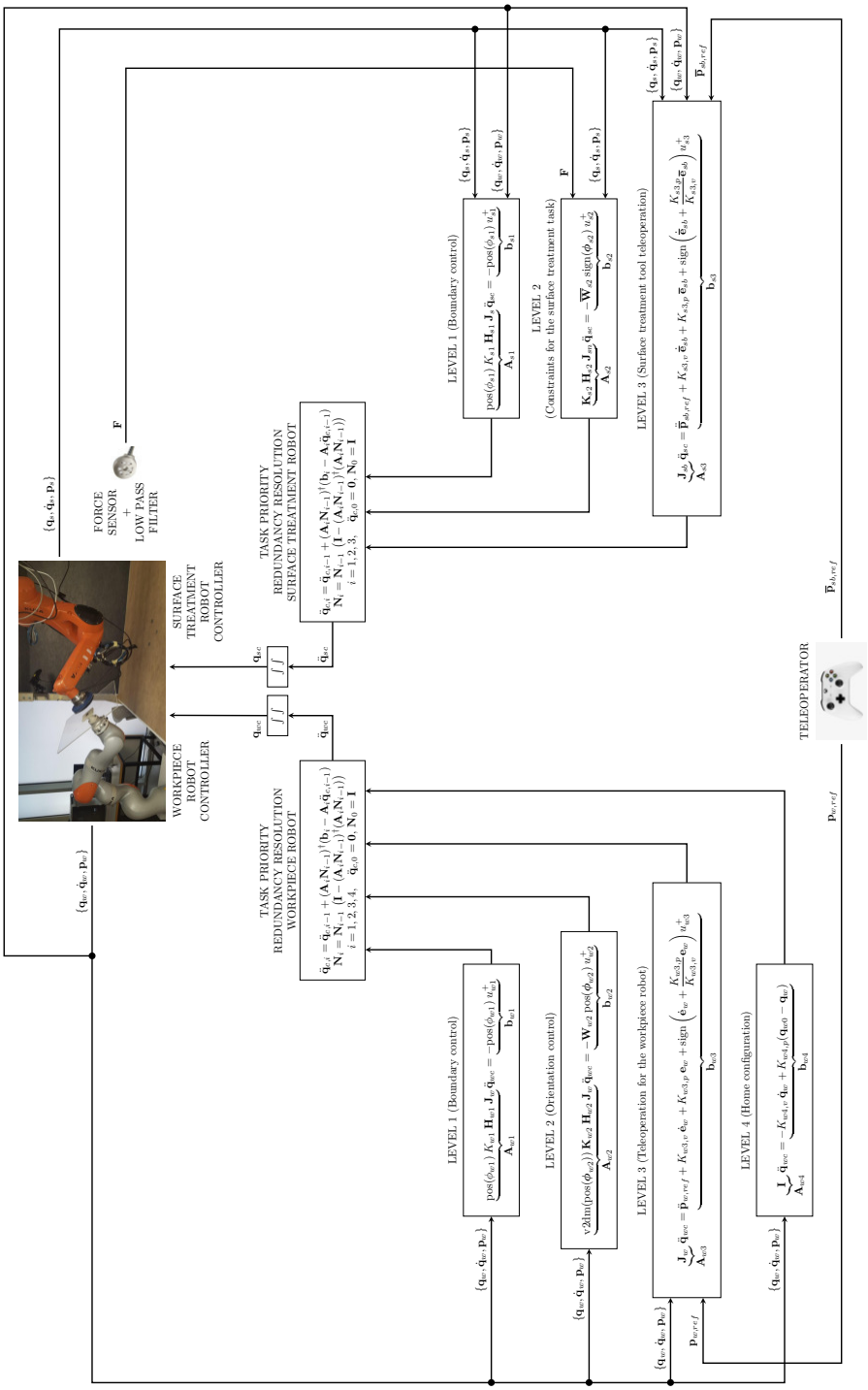


Figure 5.1: Block diagram of the proposed control for the WR and STR.

position and orientation for the task. Moreover, the teleoperator commands two linear coordinates of the STR to move the treatment tool on the workpiece surface in order to apply the surface treatment, while the STR control automatically maintains the desired pressure and perpendicularity to the surface using the data from an F/T sensor.

Furthermore, in order to assist the human user during the teleoperation, several constraints are defined for both robot arms in order to avoid exceeding the allowed workpsace. In particular, a boundary constraint is considered for the WR to confine the workpiece center inside the allowed 3D area in order to avoid collisions with other objects in the environment, see Section 5.2.5.1 for further details. Moreover, a second constraint is considered for the WR to limit the angular positions that the workpiece can achieve while the user teleoperates the WR in order to avoid an excessive tilt of the workpiece, see Section 5.2.5.2 for further details. Finally, another boundary constraint is considered for the STR to confine the STR tool inside the allowed 2D area on the workpiece surface in order to avoid exceeding the workpiece limits, see Section 5.2.6.1 for further details.

Note that other similar constraints could also be considered for both robot arms, e.g., the typical robot constraints for the maximum joint ranges. Details omitted for brevity.

5.2.2 Lie derivatives

To use the SMC in Section 3.2, a dynamical system in the form of Eq. (3.10) is considered for both WR and STR with the state vector $\mathbf{x} = \begin{bmatrix}\mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, the disturbance vector $\mathbf{d} = \mathbf{d}_c$, where $\mathbf{d}_c$ represents inaccuracies of the low-level controller developed by the robot manufacturer, and the input vector $\mathbf{u} = \ddot{\mathbf{q}}_c$. Thus, the state equation of the system, which is a double integrator, is given by:

$$\dot{\mathbf{x}} = \begin{bmatrix}\mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O}\end{bmatrix}\mathbf{x} + \begin{bmatrix}\mathbf{0} \\ \mathbf{d}_c\end{bmatrix} + \begin{bmatrix}\mathbf{O} \\ \mathbf{I}\end{bmatrix}\mathbf{u}, \tag{5.1}$$

and, hence, the Lie derivatives for the constraint function ϕ_i result in:

$$\mathbf{L}_{\mathbf{g}}\phi_i = \nabla\phi_i^{\mathrm{T}}\,\mathbf{g} = (\partial\phi_i/\partial\dot{\mathbf{q}})^{\mathrm{T}} \tag{5.2}$$

$$L_f\phi_i = \nabla\phi_i^{\mathrm{T}}\,\mathbf{f} = (\partial\phi_i/\partial\mathbf{q})^{\mathrm{T}}\;\dot{\mathbf{q}} + (\partial\phi_i/\partial\dot{\mathbf{q}})^{\mathrm{T}}\;\mathbf{d}_c. \tag{5.3}$$

5.2.3 Boundary model

The first levels of the control algorithms of both robot arms include inequality constraints required to keep the center of the workpiece and the surface treatment tool within the allowed workspace, whose boundary has the shape of a superellipsoid and a superellipse, respectively, see 3.1.3.

5.2.4 Force model

The second level of control for the STR includes several equality constraints to properly perform the treatment task. These constraints are defined in Section 5.2.6.2 depending on the vector $\mathbf{F}$ of force and torque measured by the treatment sensor. In many cases, the vector force $\mathbf{F}$ between the environment and the robotic system can be approximated by the ideal elastic model below (Siciliano et al., 2009):

$$\mathbf{F}(\mathbf{p}_s, t) = \mathbf{K}_{ts}(t)(\mathbf{p}_s(t) - \mathbf{p}_e(t)) = \begin{bmatrix} F_x & F_y & F_z & F_\alpha & F_\beta & F_\gamma \end{bmatrix}^{\mathrm{T}}, \tag{5.4}$$

where diagonal matrix $\mathbf{K}_{ts}$ is composed of the treatment sensor stiffness coefficients in each axis, $\mathbf{p}_s$ is the STR pose vector and $\mathbf{p}_e$ is the environment pose vector. Matrix $\mathbf{K}_{ts}$ and vector $\mathbf{p}_e$ are, in general, variable.

5.2.5 Control for the Workpiece robot

5.2.5.1 Level 1: Boundary control

An inequality constraint is defined based on the superellipsoid equation described in (3.7) as follows:

$$\sigma_{w1}(\mathbf{p}_w) = -1 + \left|\frac{x_{wb}}{W}\right|^m + \left|\frac{y_{wb}}{H}\right|^m + \left|\frac{z_{wb}}{M}\right|^m \leq 0, \tag{5.5}$$

where σ_{w1} defines the boundary of the allowed workspace for the workpiece center position as a superellipsoid (see Section3.1.3), $\{W, H, M\}$ and m are the axes and smoothing factor, respectively, of the superellipsoid and:

$$\overline{\mathbf{p}}_{wb}(\mathbf{p}_w) = \begin{bmatrix} x_{wb} \\ y_{wb} \\ z_{wb} \end{bmatrix} = {}^b\mathbf{R}_w \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_w - \overline{\mathbf{p}}_{wc} \right), \tag{5.6}$$

where $\overline{\mathbf{p}}_{wb}$ is the 3D position of the workpiece center relative to the boundary (i.e., the superellipsoid coordinate system), $\mathbf{p}_w$ is the WR pose vector[1], $\overline{\mathbf{p}}_{wc}$ is the 3D position of the boundary/superellipsoid center relative to the WR coordinate system, ${}^b\mathbf{R}_w$ represents the orientation of the boundary/superellipsoid (i.e., the rotation matrix that transforms WR coordinates into superellipsoid coordinates) and $\mathbf{I}_3$ and $\mathbf{O}_3$ denote the identity and zero matrix, respectively, of dimension 3×3.

Note that, as intended, in order for the inequality constraint above to be respected, the workpiece center must be kept within the limits of the superellipsoid. Otherwise, the constraint will become active.

Taking into account (3.25) and (5.5)–(5.6), the modified constraint function ϕ_{w1} for this level results in:

$$\begin{aligned}\phi_{w1}(\mathbf{p}_w) &= \sigma_{w1} + K_{w1}\,\dot{\sigma}_{w1} = \sigma_{w1} + K_{w1}\,(\partial\sigma_{w1}/\partial\mathbf{q}_w)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= \sigma_{w1} + K_{w1}\,\left((\partial\mathbf{p}_w/\partial\mathbf{q}_w)^{\mathrm{T}}\,(\partial\overline{\mathbf{p}}_{wb}/\partial\mathbf{p}_w)^{\mathrm{T}}\,(\partial\sigma_{w1}/\partial\overline{\mathbf{p}}_b)\right)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= -1 + \left|\frac{x_{wb}}{W}\right|^m + \left|\frac{y_{wb}}{H}\right|^m + \left|\frac{z_{wb}}{M}\right|^m + K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w\,\dot{\mathbf{q}}_w, \end{aligned} \tag{5.7}$$

where K_{w1} is the approaching parameter for the WR first level, $\mathbf{J}_w$ is the WR Jacobian matrix and:

$$\mathbf{H}_{w1} = m \begin{bmatrix} \dfrac{\mathrm{sign}(x_{wb})\ |x_{wb}|^{(m-1)}}{W^m} \\ \dfrac{\mathrm{sign}(y_{wb})\ |y_{wb}|^{(m-1)}}{H^m} \\ \dfrac{\mathrm{sign}(z_{wb})\ |z_{wb}|^{(m-1)}}{M^m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} {}^b\mathbf{R}_w & \mathbf{O}_3 \end{bmatrix}. \tag{5.8}$$

Furthermore, from (5.2) and (5.7), the required Lie derivative $\mathbf{L_g}\phi_1$ results in:

$$\mathbf{L_g}\phi_{w1} = (\partial\phi_{w1}/\partial\dot{\mathbf{q}}_w)^{\mathrm{T}} = K_{w1}(\partial\sigma_{w1}/\partial\mathbf{q}_w)^{\mathrm{T}} = K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w. \tag{5.9}$$

From Eqs. (3.15) and (5.9), the control equation for WR Level 1 results in:

$$\mathrm{pos}(\phi_{w1})\,K_{w1}\,\mathbf{H}_{w1}\,\mathbf{J}_w\,\ddot{\mathbf{q}}_{wc} = -\mathrm{pos}(\phi_{w1})\,u^+_{w1} \;\rightarrow\; \mathbf{A}_{w1}\,\ddot{\mathbf{q}}_{wc} = b_{w1}, \tag{5.10}$$

[1]Note that it has been assumed that the WR grasps the workpiece at its center and, hence, the workpiece center position corresponds to the first three components of the WR pose vector $\mathbf{p}_w$. However, if that would not be the case, the formulation can be easily modified, details omitted for brevity.

where u_{w1}^{+} represents the switching gain of the SMC and $\mathbf{A}_{w1}$ and b_{w1} correspond to the matrix and vector (a row vector and a scalar in this case), respectively, for the WR first task in (3.4).

5.2.5.2 Level 2: Orientation control

In order to keep the orientation of the workpiece around a reference value, three inequality constraints are defined as follows:

$$\sigma_{w2,\alpha}(\mathbf{p}_w) = |\alpha_w - \alpha_{w,mid}| - \alpha_{w,max} \leq 0 \tag{5.11}$$
$$\sigma_{w2,\beta}(\mathbf{p}_w) = |\beta_w - \beta_{w,mid}| - \beta_{w,max} \leq 0 \tag{5.12}$$
$$\sigma_{w2,\gamma}(\mathbf{p}_w) = |\gamma_w - \gamma_{w,mid}| - \gamma_{w,max} \leq 0, \tag{5.13}$$

where $\{\alpha_w, \beta_w, \gamma_w\}$ are the actual values for the WR roll, pitch and yaw orientation angles, $\{\alpha_{w,mid}, \beta_{w,mid}, \gamma_{w,mid}\}$ are the mid-range values for these angles and $\{\alpha_{w,max}, \beta_{w,max}, \gamma_{w,max}\}$ are the maximum absolute allowed deviation angles with respect to the mid-range values.

Taking into account (3.25) and (5.11)–(5.13), the modified constraint function vector $\boldsymbol{\phi}_{w2}$ for this level results in:

$$\begin{aligned}\boldsymbol{\phi}_{w2}(\mathbf{p}_w) &= \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,\dot{\boldsymbol{\sigma}}_{w2} = \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{q}_w)^{\mathrm{T}}\,\dot{\mathbf{q}}_w \\ &= \boldsymbol{\sigma}_{w2} + \mathbf{K}_{w2}\,(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{p}_w)^{\mathrm{T}}(\partial\mathbf{p}_w/\partial\mathbf{q}_w)\,\dot{\mathbf{q}}_w \\ &= \begin{bmatrix} |\alpha_w - \alpha_{w,mid}| - \alpha_{w,max} \\ |\beta_w - \beta_{w,mid}| - \beta_{w,max} \\ |\gamma_w - \gamma_{w,mid}| - \gamma_{w,max} \end{bmatrix} + \mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_w\,\dot{\mathbf{q}}_w,\end{aligned} \tag{5.14}$$

where $\boldsymbol{\phi}_{w2}$ and $\boldsymbol{\sigma}_{w2}$ are column vectors composed of the modified and original constraint functions $\phi_{w2,i}$ and $\sigma_{w2,i}$, respectively, see (5.11)-(5.13), $\mathbf{K}_{w2}$ is a diagonal matrix composed of all approaching parameters for the WR second level, $\mathbf{J}_w$ is the WR Jacobian matrix and:

$$\mathbf{H}_{w2} = \begin{bmatrix} 0 & 0 & 0 & \mathrm{sign}(\alpha_w - \alpha_{w,mid}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{sign}(\beta_w - \beta w, ref) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathrm{sign}(\gamma_w - \gamma w, ref) \end{bmatrix}. \tag{5.15}$$

Furthermore, from (5.2) and (5.14), the required Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{w2}$ results in:

$$\mathbf{L_g}\boldsymbol{\phi}_{w2} = (\partial\boldsymbol{\phi}_{w2}/\partial\dot{\mathbf{q}}_w)^{\mathrm{T}} = \mathbf{K}_{w2}(\partial\boldsymbol{\sigma}_{w2}/\partial\mathbf{q}_w)^{\mathrm{T}} = \mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_w. \tag{5.16}$$

Thus, the control equality for the WR second level is:

$$\text{v2dm}(\text{pos}(\boldsymbol{\phi}_{w2}))\,\mathbf{K}_{w2}\,\mathbf{H}_{w2}\,\mathbf{J}_{w}\,\ddot{\mathbf{q}}_{wc} = -\mathbf{W}_{w2}\,\text{pos}(\boldsymbol{\phi}_{w2})\,u^{+}_{w2} \;\rightarrow\; \mathbf{A}_{w2}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w2}, \tag{5.17}$$

where u^{+}_{w2} is the switching gain, $\mathbf{W}_{w2}$ is the switching gain weight matrix, and $\mathbf{b}_{w2}$ and $\mathbf{A}_{w2}$ are the vector and matrix, respectively, for the WR third task in (3.4).

5.2.5.3 Level 3: Teleoperation for the workpiece robot

In order to add flexibility to the surface treatment operation, the third control level of the WR allows the user to teleoperate the position and orientation of the workpiece.

Taking into account (3.3), the following control equation is obtained to achieve this teleoperation:

$$\begin{aligned}\mathbf{J}_{w}\ddot{\mathbf{q}}_{wc} &= \ddot{\mathbf{p}}_{w,ref} + K_{w3,v}\,\dot{\mathbf{e}}_{w} + K_{w3,p}\,\mathbf{e}_{w} + \text{sign}\left(\dot{\mathbf{e}}_{w} + \frac{K_{w3,p}}{K_{w3,v}}\mathbf{e}_{w}\right) u^{+}_{w3} \\ &\rightarrow\; \mathbf{A}_{w3}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w3},\end{aligned} \tag{5.18}$$

where $\mathbf{J}_{w}$ is the WR Jacobian matrix, $\mathbf{e}_{w} = \mathbf{p}_{w,ref} - \mathbf{p}_{w}$ and $\dot{\mathbf{e}}_{w} = \dot{\mathbf{p}}_{w,ref} - \dot{\mathbf{p}}_{w}$ represent the WR pose and velocity error, respectively, $K_{w3,p}$ and $K_{w3,v}$ are their correspondent correction gains, and $\mathbf{b}_{w3}$ and $\mathbf{A}_{w3}$ are the vector and matrix for the WR third task in (3.4). It must be noted that $\dot{\mathbf{p}}_{w}$ is obtained from the first order robot kinematics in (3.2), and that the reference $\mathbf{p}_{w,ref}$ is generated by the human teleoperator.

Note that a hybrid control approach is developed for this control level, since the term $\dot{\mathbf{J}}_{w}\dot{\mathbf{q}}_{w}$ from Eq. (3.3) is substituted by the conventional SMC switching term $\text{sign}\left(\dot{\mathbf{e}}_{w} + \frac{K_{w3,p}}{K_{w3,v}}\mathbf{e}_{w}\right)$. This hybrid control presents two advantages: the Jacobian derivative is not needed, so its computation is avoided; and, thanks to the continuous control terms in the control action, the switching gain $\mathbf{u}^{+}_{w3}$ can be relatively small, achieving a reduced chattering effect.

5.2.5.4 Level 4: Home configuration

If the STR robot is *redundant*, e.g., like the 7R cobot used in the experiments below, the DoF that remain at this point can be used for other objectives,

avoiding also a bias self-motion. In this research, a home configuration $\mathbf{q}_{w0}$ is considered to "push" the STR to it for increasing safety. In particular, the following control law is used:

$$\ddot{\mathbf{q}}_{wc} = -K_{w4,v}\,\dot{\mathbf{q}}_w + K_{w4,p}(\mathbf{q}_{w0} - \mathbf{q}_w) \;\rightarrow\; \mathbf{A}_{w4}\,\ddot{\mathbf{q}}_{wc} = \mathbf{b}_{w4}, \tag{5.19}$$

where $K_{w4,v}$ and $K_{w4,p}$ are the velocity and position correction gains, respectively, and $\mathbf{b}_{w4}$ and $\mathbf{A}_{w4}$ are the vector and matrix for the WR fourth task in (3.4)

5.2.6 Control for the surface treatment robot

5.2.6.1 Level 1: Boundary control

Similarly to Level 1 of the WR control (see Section 5.2.5.1), an inequality constraint is defined based on the modified superellipse equation described in (3.9) as follows:

$$\sigma_{s1}(\mathbf{p}_s, t) = -1 + \left|\frac{x_{sb}}{W}\right|^m + \left(\frac{\max(|y_{sb}| - (H - W), 0)}{W}\right)^m \leq 0, \tag{5.20}$$

where σ_{s1} defines the boundary of the allowed surface on the workpiece for the STR tool as a modified superellipse (see Section3.1.3), $\{W, H\}$ and m are the axes and smoothing factor, respectively, of the modified superellipse and:

$$\overline{\mathbf{p}}_{sb}(\mathbf{p}_s, t) = \begin{bmatrix} x_{sb} \\ y_{sb} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^b\mathbf{R}_s(t) \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_s - \overline{\mathbf{p}}_{sc}(t) \right), \tag{5.21}$$

where $\overline{\mathbf{p}}_{sb}$ is the 2D position of the STR tool with respect to the boundary (i.e., the superellipse coordinate system), $\mathbf{p}_s$ is the STR pose vector, $\overline{\mathbf{p}}_{sc}$ is the 3D position of the superellipse center relative to the STR coordinate system, ${}^b\mathbf{R}_s$ represents the orientation of the boundary/superellipse (i.e., the rotation matrix that transforms STR coordinates into superellipse coordinates) and $\mathbf{I}_3$ and $\mathbf{O}_3$ denote the identity and zero matrix, respectively, of dimension 3×3.

Note that the values of the position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^b\mathbf{R}_s$ of the modified superellipse are readily obtained from the WR pose vector $\mathbf{p}_w$, since the WR holds the workpiece and the superellipse is virtually attached to the workpiece surface.

As before, in order for the inequality constraint above to be respected, the STR tool position must be kept within the limits of the modified superellipse. Otherwise, the constraint will become active.

Taking into account (3.25), the modified constraint function ϕ_{s1} for this level results in:

$$\phi_{s1}(\mathbf{p}_s, t) = \sigma_{s1} + K_{s1}\,\dot{\sigma}_{s1} \tag{5.22}$$

where K_{s1} is the approaching parameter for the STR first level.

Note that the modified superellispe is non-static due to the workpiece movement, i.e., both its position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^{b}\mathbf{R}_{s}$ are variable. Therefore, the value of $\dot{\sigma}_{s1}$ in (5.22) cannot be computed from the STR joint velocity vector $\dot{\mathbf{q}}_s$ alone and, hence, numerical differentiation of σ_{s1} is used in the practical implementation below.

Taking into account (5.21) and (5.22), the use of a modified superellipse as described in (3.9) is justified as follows. If $H > W$ and σ_{s1} was based on the regular superellipse given by (3.8), a movement in the X-axis of the superellipse would cause a much faster increase in σ_{s1} and ϕ_{s1} than a movement in the Y-axis with the same speed, which would, in turn, cause the control action to activate sooner in the first case and, thus, make the approach to the boundary smoother than in the second case, where the control action would be later and, as a consequence, rougher.

In contrast, by using the modified superellipse equation (3.9), the allowed workspace remains a $2W \times 2H$ rectangle with smoothed corners, see Fig. 3.1, while the way in which σ_{s1} and ϕ_{s1} evolve depending on the direction of the tool movement is more homogeneous, since both terms $|x_{sb}|$ and $\max(|y_{sb}| - (H - W), 0)$ are bounded between 0 and W.

From (5.2), (5.22) and (5.21), the required Lie derivative $\mathbf{L_g}\phi_{s1}$ results in:

$$\begin{aligned}
\mathbf{L_g}\phi_{s1} &= (\partial\phi_{s1}/\partial\dot{\mathbf{q}}_s)^{\mathrm{T}} = K_{s1}\,(\partial\sigma_{s1}/\partial\mathbf{q}_s)^{\mathrm{T}} \\
&= K_{s1}\,\left((\partial\mathbf{p}_s/\partial\mathbf{q}_s)^{\mathrm{T}}\,(\partial\overline{\mathbf{p}}_{sb}/\partial\mathbf{p}_s)^{\mathrm{T}}\,(\partial\sigma_{s1}/\partial\overline{\mathbf{p}}_s)\right)^{\mathrm{T}} \\
&= K_{s1}\,\mathbf{H}_{s1}\,\mathbf{J}_s,
\end{aligned} \tag{5.23}$$

where $\mathbf{J}_s$ is the STR Jacobian matrix and:

$$\mathbf{H}_{s1} = m \begin{bmatrix} \dfrac{\mathrm{sign}(x_{sb})\ |x_{sb}|^{(m-1)}}{W^m} \\ \dfrac{(\max(|y_{sb}| - (H - W), 0))^{(m-1)}}{H^m} \\ 0 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} {}^{b}\mathbf{R}_s & \mathbf{O}_3 \end{bmatrix}. \tag{5.24}$$

From (3.15) and (5.23), the control equation for STR Level 1 results in:

$$\text{pos}(\phi_{s1})\, K_{s1}\, \mathbf{H}_{s1}\, \mathbf{J}_s\, \ddot{\mathbf{q}}_{sc} = -\text{pos}(\phi_{s1})\, u_{s1}^{+} \;\rightarrow\; \mathbf{A}_{s1}\, \ddot{\mathbf{q}}_{sc} = b_{s1}, \tag{5.25}$$

where u_{s1}^{+} represents the switching gain of the SMC and $\mathbf{A}_{s1}$ and b_{s1} correspond to the matrix and vector (a row vector and a scalar in this case), respectively, for the STR first task in (3.4).

5.2.6.2 Level 2: Treatment task constraints

The following equality constraints are considered for the robot surface treatment:

$$\sigma_{s2,z}(\mathbf{F}) = \sigma_{s2,z}(\mathbf{p}_s, t) = F_z - F_{z,ref} = 0 \tag{5.26}$$
$$\sigma_{s2,\alpha}(\mathbf{F}) = \sigma_{s2,\alpha}(\mathbf{p}_s, t) = F_\alpha = 0 \tag{5.27}$$
$$\sigma_{s2,\beta}(\mathbf{F}) = \sigma_{s2,\beta}(\mathbf{p}_s, t) = F_\beta = 0, \tag{5.28}$$

where F_z is the measurement of the F/T sensor in the linear Z-axis of STR tool, F_α and F_β are the measurements of the F/T sensor in angular X- and Y-axes of the STR tool, and $F_{z,ref}$ is the reference value for the force F_z. Thus, the first equality above is used to accomplish the reference pressure $F_{z,ref}$ between the workpiece and the STR tool, and the last two equalities are used to maintain the tool orthogonality to the workpiece, i.e., the above angular torques are zero if the STR tool is completely orthogonal to the workpiece. Note that the human operator could teleoperate the torque in the angular Z-axis because it is not restricted.

From (3.24) and (5.26)–(5.28), the modified constraint function vector $\boldsymbol{\phi}_{s2}$ for this level results in:

$$\boldsymbol{\phi}_{s2}(\mathbf{F}, \dot{\mathbf{F}}) = \boldsymbol{\sigma}_{s2} + \mathbf{K}_{s2}\, \dot{\boldsymbol{\sigma}}_{s2} = \begin{bmatrix} F_z - F_{z,ref} + K_{s2,z}\, \dot{F}_z \\ F_\alpha + K_{s2,\alpha}\, \dot{F}_\alpha \\ F_\beta + K_{s2,\beta}\, \dot{F}_\beta \end{bmatrix}, \tag{5.29}$$

where $\boldsymbol{\phi}_{s2}$ and $\boldsymbol{\sigma}_{s2}$ are column vectors composed of the modified and original constraint functions $\phi_{s2,i}$ and $\sigma_{s2,i}$, respectively, and diagonal matrix $\mathbf{K}_{s2}$ with the approaching parameters for the STR second level.

Furthermore, taking into account (3.24), (5.2), (5.4) and (5.26) –(5.28),

the required Lie derivative $\mathbf{L_g}\boldsymbol{\phi}_{s2}$ results in:

$$\mathbf{L_g}\boldsymbol{\phi}_{s2} = (\partial\boldsymbol{\phi}_{s2}/\partial\dot{\mathbf{q}}_s)^{\mathrm{T}} = \mathbf{K}_{s2}(\partial\boldsymbol{\sigma}_{s2}/\partial\mathbf{q}_s)^{\mathrm{T}} = \mathbf{K}_{s2}(\partial\boldsymbol{\sigma}_{s2}/\partial\mathbf{p}_s)^{\mathrm{T}}(\partial\mathbf{p}_s/\partial\mathbf{q}_s)$$
$$= \mathbf{K}_{s2}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}\mathbf{K}_{ts}\mathbf{J}_{sn} = \mathbf{K}_{s2}\mathbf{H}_{s2}\mathbf{K}_{ts}\mathbf{J}_{sn}, \tag{5.30}$$

where diagonal matrix $\mathbf{K}_{ts}$ is associated to the treatment sensor stiffness and $\mathbf{J}_{sn}$ is the STR geometric Jacobian relative to the tool coordinate system (Siciliano et al., 2009), that is, the Jacobian that relates the velocity vector $\dot{\mathbf{q}}_s$ and the STR end-effector velocities with respect to the treatment tool frame.

The stiffness parameters $\mathbf{K}_{ts}$ in $\mathbf{L_g}\boldsymbol{\phi}_{s2}$ (5.30) are usually not known, but, without loss of generality, they can be incorporated in the switching gain weight matrix $\mathbf{W}_{s2}$. Thus, the SMC given by (3.12) results in:

$$\mathbf{K}_{s2}\,\mathbf{H}_{s2}\,\mathbf{J}_{sn}\,\ddot{\mathbf{q}}_{sc} = -\overline{\mathbf{W}}_{s2}\,\mathrm{sign}(\boldsymbol{\phi}_{s2})\,u_{s2}^{+} \;\rightarrow\; \mathbf{A}_{s2}\,\ddot{\mathbf{q}}_{sc} = \mathbf{b}_{s2}, \tag{5.31}$$

where u_{s2}^{+} represents the switching gain of the SMC and $\mathbf{b}_{s2}$ and $\mathbf{A}_{s2}$ are the vector and matrix for the STR second task in (3.4) and:

$$\overline{\mathbf{W}}_{s2} = \begin{bmatrix} W_{s2,z}/K_{ts,z} & 0 & 0 \\ 0 & W_{s2,\alpha}/K_{ts,\alpha} & 0 \\ 0 & 0 & W_{s2,\beta}/K_{ts,\beta} \end{bmatrix} = \begin{bmatrix} \overline{W}_{s2,z} & 0 & 0 \\ 0 & \overline{W}_{s2,\alpha} & 0 \\ 0 & 0 & \overline{W}_{s2,\beta} \end{bmatrix}. \tag{5.32}$$

Note that the conventional SMC given by (5.31) only requires: the control parameters $\{u_{s2}^{+}, \overline{\mathbf{W}}_{s2}, \mathbf{K}_{s2}, F_{z,ref}\}$; the robot Jacobian $\mathbf{J}_{sn}$; and the treatment sensor measurement $\mathbf{F}$ and its derivative.

5.2.6.3 Level 3: Surface treatment tool teleoperation

In order to carry out the surface treatment operation, the third control level of the STR allows the user to teleoperate the tool position on the workpiece surface with respect to the superellipse coordinate system.

Similarly to Level 3 of the WR control (see Sec. 5.2.5.3), the following control equation is obtained to achieve this teleoperation:

$$\mathbf{J}_{sb}\ddot{\mathbf{q}}_{sc} = \ddot{\overline{\mathbf{p}}}_{sb,ref} + K_{s3,v}\,\dot{\overline{\mathbf{e}}}_{sb} + K_{s3,p}\,\overline{\mathbf{e}}_{sb} + \mathrm{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \frac{K_{s3,p}}{K_{s3,v}}\overline{\mathbf{e}}_{sb}\right)u_{s3}^{+}$$
$$\rightarrow \quad \mathbf{A}_{s3}\,\ddot{\mathbf{q}}_{sc} = \mathbf{b}_{s3}, \tag{5.33}$$

where $\overline{\mathbf{p}}_{sb,ref}$ is the reference generated by the human teleoperator for the 2D position of the STR tool with respect to the boundary/superellipse coordinate system, $K_{s3,p}$ and $K_{s3,v}$ are the position and velocity correction gains, respectively, u_{s3}^{+} represents the switching gain of the SMC, $\mathbf{b}_{s3}$ and $\mathbf{A}_{s3}$ are the vector and matrix for the STR third task in (3.4) and:

$$\mathbf{J}_{sb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^{b}\mathbf{R}_{s} & \mathbf{O}_{3} \end{bmatrix} \mathbf{J}_{s} \tag{5.34}$$

$$\overline{\mathbf{e}}_{sb} = \overline{\mathbf{p}}_{sb,ref} - \overline{\mathbf{p}}_{sb} \tag{5.35}$$

$$\dot{\overline{\mathbf{e}}}_{sb} = \dot{\overline{\mathbf{p}}}_{sb,ref} - \dot{\overline{\mathbf{p}}}_{sb} \tag{5.36}$$

where $\overline{\mathbf{e}}_{sb}$ and $\dot{\overline{\mathbf{e}}}_{sb}$ are the 2D position and velocity errors, respectively, of the STR tool with respect to the boundary/superellipse coordinate system.

Note that the derivative $\dot{\overline{\mathbf{p}}}_{sb}$ in (5.36) cannot be computed from the STR joint velocity vector $\dot{\mathbf{q}}_{s}$ alone since, as mentioned above, the boundary/superellipse is non-static due to the workpiece movement, see (5.21). Hence, numerical differentiation of $\overline{\mathbf{p}}_{sb}$ is used in the practical implementation below.

In the same way as in Level 3 for the WR control, note that a hybrid control approach is developed for this control level, substituting $\dot{\mathbf{J}}_{sb}\dot{\mathbf{q}}_{s}$ by the conventional SMC switching term $\operatorname{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \dfrac{K_{s3,p}}{K_{s3,v}}\overline{\mathbf{e}}_{sb}\right)$.

5.2.7 Limitations of the proposed approach

The main limitations of the proposed method are discussed as follows:

– *High-level planning*: The proposed robot control does not include high-level planning, i.e., the robot controller "reacts" to the teleoperator commands and robot constraints with no kind of prediction or long-term anticipation. Therefore, the algorithm may be blocked in trap situations (Gracia et al., 2012) or achieve singular configurations (Gracia et al., 2009) where the robot system "loses" DoF, which may degrade the performance of the robot task. In some cases, these situations could be avoided by employing a high-level planner (see the discussion about task optimization in Section 2.4) to solve the problem using a large prediction horizon and taking into account the complete data, including a priori knowledge of the teleoperator commands, which may not be possible in practice. Nevertheless, the complexity and computational cost

of this high-level planner would be significantly greater than those of the proposed method.

However, singular configurations could also be avoided in the proposed method by including, similarly to the boundary constraints above, another inequality constraint in order to prevent the robot from entering workspace regions where the condition number of the robot Jacobian is above a given threshold (note that, at singular configurations, the condition number tends to infinity). However, this is out of the scope of this research and remains as further work.

– *Chattering*: SMC discrete-time implementations make the system oscillate with finite amplitude and frequency within a band around $\boldsymbol{\phi} = \mathbf{0}$, namely *chattering* (Edwards and Spurgeon, 1998). For the proposed method, the chattering band $\triangle\boldsymbol{\phi}$ has an upper bound that can be computed as follows using the Euler-integration of the SMC action in (3.12):

$$\triangle\boldsymbol{\phi} = T\,|\mathbf{L_g}\boldsymbol{\phi}\,\mathbf{u}| = T\,u^{+}\,\mathrm{dm2v}(\mathbf{W}), \tag{5.37}$$

where function $\mathrm{dm2v}(\cdot)$ converts a diagonal matrix into a column vector and T^{-1} is the sampling frequency of the robotic system.

Nevertheless, for reasonably fast sampling the chattering drawback can be neglected. For instance, in the real experimentation of Section 5.4, sampling frequencies of 75Hz and 200Hz gave rise to negligible chattering effect.

However, it is important to remark that there are several approaches in the literature to reduce the chattering effect of the SMC: using a super-twisting or twisting SMC (Levant, 2003); using a quasi-continuous SMC (Levant, 2005); using the hyperbolic tangent curve as switching function (Edwards and Spurgeon, 1998); etc. However, this is out of the scope of this research.

5.3 Controller implementation

Table 5.1 and Table 5.2 show the pseudo-code of the controllers developed in this work for the WR and STR, respectively. Note that it is assumed that these controllers are implemented in a modern computer, although they could also be easily implemented in other platforms such as: an industrial workstation;

an embedded processor; a Programmable Logic Controller (PLC); or even directly in the robot controller.

The algorithms of the WR and STR control are executed at T_w and T_s seconds sampling periods, respectively, and incorporate the auxiliary functions below:

- Kinematic functions and Jacobian matrices: $\mathbf{l}_w(\mathbf{q}_w)$, $\mathbf{l}_s(\mathbf{q}_s)$, $\mathbf{J}_w(\mathbf{q}_w)$, $\mathbf{J}_s(\mathbf{q}_s)$ and $\mathbf{J}_{sn}(\mathbf{q}_s)$.
- Moore-Penrose pseudoinverse (a tolerance is used to discard small singular values, see Section 3.1): $(\cdot)^\dagger$.
- Function *Pose2PositionAndRot*($\mathbf{p}_w$) that converts the WR pose vector into the position $\overline{\mathbf{p}}_{sc}$ and orientation ${}^b\mathbf{R}_s$ of the modified superellipse.
- Robot sensors:
  - *GetWRState*() for the WR algorithm, which returns the current WR state $\{\mathbf{q}_w, \dot{\mathbf{q}}_w\}$.
  - *GetSTRStateAndForces*() for the STR algorithm, which returns the current STR state $\{\mathbf{q}_s, \dot{\mathbf{q}}_s\}$ and forces $\mathbf{F}$ measured by the treatment sensor, which is assumed that has been filtered by the electronics of the treatment sensor.
- Robot communications:
  - *GetUserReferenceWR*() which reads the reference pose $\mathbf{p}_{w,ref}$ determined by the human teleoperator for the workpiece, in the WR algorithm.
  - *GetUserReferenceSTR*() which reads the reference position $\overline{\mathbf{p}}_{sb,ref}$ determined by the human teleoperator for the STR tool.
  - *GetWRPose*() which obtains the current WR pose $\mathbf{p}_w$ by reading data received from the WR, in the STR algorithm.
- Actuators:
  - *SendToWRJointControllers*($\mathbf{q}_{wc}$), which commands the desired joint values to the WR joint controllers.
  - *SendToSTRJointControllers*($\mathbf{q}_{sc}$), which commands the desired joint values to the STR joint controllers.

Note that, in the code of Table 5.1 and Table 5.2, numerical differentiation (i.e., the well-known backward Euler approximation) is used to obtain the derivatives of the signals $\{\mathbf{p}_{w,ref}\}$ and $\{\sigma_{s1}, \mathbf{F}, \overline{\mathbf{p}}_{sb}, \overline{\mathbf{p}}_{sb,ref}\}$ in the WR and STR control algorithms, respectively. However, depending on the application, this approach could lead to excessive noise in the signals. In order to avoid this issue, the sampling period of the corresponding control algorithm should be chosen large enough in order to avoid noisy signals[2]. For instance, in the specific case of the experimentation in Section 5.4, a synchronous sampling period of 5 ms and 13 ms for the WR and STR control algorithms, respectively, gave rise to negligible noise in numerical differentiation.

From a computational complexity point of view, the algorithms of the WR and STR have 31 and 35 lines of code, respectively, see Table 5.1 and Table 5.2. Moreover, the computational cost of both algorithms implemented in a modern computer is around 12 microseconds for the experiments in Section 5.4, although this value could change if the algorithms are implemented in a different platform, e.g., an industrial workstation, a PLC, etc.

5.4 Real experimentation

5.4.1 Setup

The experimental platform of this work can be seen in Fig. 5.2. Its components are detailed below.

– STR: A 6R robot arm (KUKA KR6 Agilus), whose Denavit-Hartenberg parameters are displayed in Table 5.3; an Axia80 F/T sensor attached to the STR end-effector; and a self-developed brush of 14 cm in diameter to carry out a demonstrative surface treatment, which is attached to the F/T sensor.

– WR: A 7R cobot (KUKA LBR iiwa 14 r820), whose DH parameters are shown in Table 5.4; and a methacrylate board measuring 28x38 cm (the workpiece) rigidly attached to the WR end-effector by means of a self developed adaptor.

[2]Note that if a small sampling period is chosen, a low-pass filter would be required to remove the noise from the derivatives. However, the bandwidth of this approach is approximately equivalent to use a larger sampling period with no filtering.

Table 5.1: Code of the algorithm

Algorithm executed at sampling time of T_w seconds	
$[\mathbf{q}_w, \dot{\mathbf{q}}_w]$ =`GetWRState();`	
$\mathbf{p}_{w,ref}$ =`GetUserReferenceWR();`	
$\mathbf{p}_w = \mathbf{l}_w(\mathbf{q}_w)$;	// Eq. (3.1)
$\dot{\mathbf{p}}_w = \mathbf{J}_w \dot{\mathbf{q}}_w$;	// Eq. (3.2)
$\dot{\mathbf{p}}_{w,ref} = (\mathbf{p}_{w,ref} - \mathbf{p}_{w,ref,prev})/T_w$;	// Derivative
$\ddot{\mathbf{p}}_{w,ref} = (\dot{\mathbf{p}}_{w,ref} - \dot{\mathbf{p}}_{w,ref,prev})/T_w$;	// Derivative
$\overline{\mathbf{p}}_{wb} = {}^b\mathbf{R}_w \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_w - \overline{\mathbf{p}}_{wc} \right)$;	// Eq. (5.6)
$\phi_{w1} = -1 + \left\lvert \frac{x_{wb}}{W} \right\rvert^m + \left\lvert \frac{y_{wb}}{H} \right\rvert^m + \left\lvert \frac{z_{wb}}{M} \right\rvert^m + K_{w1}\, \mathbf{H}_{w1}\, \mathbf{J}_w\, \dot{\mathbf{q}}_w$;	// Eq. (5.7)
$\boldsymbol{\phi}_{w2} = \begin{bmatrix} \lvert\alpha_w - \alpha_{w,mid}\rvert - \alpha_{w,max} \\ \lvert\beta_w - \beta_{w,mid}\rvert - \beta_{w,max} \\ \lvert\gamma_w - \gamma_{w,mid}\rvert - \gamma_{w,max} \end{bmatrix} + \mathbf{K}_{w2}\, \mathbf{H}_{w2}\, \mathbf{J}_w\, \dot{\mathbf{q}}_w$;	// Eq. (5.14)
$\mathbf{A}_{w1} = \mathrm{pos}(\phi_{w1})\, K_{w1}\, \mathbf{H}_{w1}\, \mathbf{J}_w$;	// Eq. (5.10)
$b_{w1} = -\mathrm{pos}(\phi_{w1})\, u^+_{w1}$;	// Eq. (5.10)
$\mathbf{A}_{w2} = \mathrm{v2dm}(\mathrm{pos}(\boldsymbol{\phi}_{w2}))\, \mathbf{K}_{w2}\, \mathbf{H}_{w2}\, \mathbf{J}_w$;	// Eq. (5.17)
$\mathbf{b}_{w2} = -\mathbf{W}_{w2}\, \mathrm{pos}(\boldsymbol{\phi}_{w2})\, u^+_{w2}$;	// Eq. (5.17)
$\mathbf{A}_{w3} = \mathbf{J}_w$;	// Eq. (5.18)
$\mathbf{b}_{w3} = \ddot{\mathbf{p}}_{w,ref} + K_{w3,v}\, \dot{\mathbf{e}}_w + K_{w3,p}\, \mathbf{e}_w + \mathrm{sign}\left(\dot{\mathbf{e}}_w + \frac{K_{w3,p}}{K_{w3,v}} \mathbf{e}_w \right) \mathbf{u}^+_{w3}$;	// Eq. (5.18)
$\mathbf{A}_{w4} = \mathbf{I}$;	// Eq. (5.19)
$\mathbf{b}_{w4} = -K_{w4,v}\, \dot{\mathbf{q}}_w + K_{w4,p}(\mathbf{q}_{w0} - \mathbf{q}_w)$;	// Eq. (5.19)
$\ddot{\mathbf{q}}_{wc,1} = \mathbf{A}^\dagger_{w1} b_{w1}$;	// Eq. (3.5), $i = 1$
$\mathbf{N}_{w1} = \mathbf{I} - \mathbf{A}^\dagger_{w1} \mathbf{A}_{w1}$;	// Eq. (3.6), $i = 1$
$\ddot{\mathbf{q}}_{wc,2} = \ddot{\mathbf{q}}_{wc,1} + (\mathbf{A}_{w2}\mathbf{N}_{w1})^\dagger (\mathbf{b}_{w2} - \mathbf{A}_{w2}\ddot{\mathbf{q}}_{wc,1})$;	// Eq. (3.5), $i = 2$
$\mathbf{N}_{w2} = \mathbf{N}_{w1}(\mathbf{I} - (\mathbf{A}_{w2}\mathbf{N}_{w1})^\dagger (\mathbf{A}_{w2}\mathbf{N}_{w1}))$;	// Eq. (3.6), $i = 2$
$\ddot{\mathbf{q}}_{wc,3} = \ddot{\mathbf{q}}_{wc,2} + (\mathbf{A}_{w3}\mathbf{N}_{w2})^\dagger (\mathbf{b}_{w3} - \mathbf{A}_{w3}\ddot{\mathbf{q}}_{wc,2})$;	// Eq. (3.5), $i = 3$
$\mathbf{N}_{w3} = \mathbf{N}_{w2}(\mathbf{I} - (\mathbf{A}_{w3}\mathbf{N}_{w2})^\dagger (\mathbf{A}_{w3}\mathbf{N}_{w2}))$;	// Eq. (3.6), $i = 3$
$\ddot{\mathbf{q}}_{wc,4} = \ddot{\mathbf{q}}_{wc,3} + (\mathbf{A}_{w4}\mathbf{N}_{w3})^\dagger (\mathbf{b}_{w4} - \mathbf{A}_{w4}\ddot{\mathbf{q}}_{wc,3})$;	// Eq. (3.5), $i = 4$
$\dot{\mathbf{q}}_{wc} = \ddot{\mathbf{q}}_{wc,4}\, T_w + \dot{\mathbf{q}}_{wc,prev}$;	// Integration
$\mathbf{q}_{wc} = \dot{\mathbf{q}}_{wc}\, T_w + \mathbf{q}_{wc,prev}$;	// Integration
`SendToWRJointControllers(`$\mathbf{q}_{wc}$`);`	
$\mathbf{q}_{wc,prev} = \mathbf{q}_{wc}$;	// For next iteration
$\dot{\mathbf{q}}_{wc,prev} = \dot{\mathbf{q}}_{wc}$;	// For next iteration
$\mathbf{p}_{w,ref,prev} = \mathbf{p}_{w,ref}$;	// For next iteration
$\dot{\mathbf{p}}_{w,ref,prev} = \dot{\mathbf{p}}_{w,ref}$;	// For next iteration

Table 5.2: Code of the algorithm

Algorithm executed at sampling time of T_s seconds
$[\mathbf{q}_s, \dot{\mathbf{q}}_s, \mathbf{F}]$ =`GetSTRStateAndForces();`
$\overline{\mathbf{p}}_{sb,ref}$ =`GetUserReferenceSTR();`
$\mathbf{p}_w$ =`GetWRPose();`
$[\overline{\mathbf{p}}_{sc},\ {}^b\mathbf{R}_s]$ =`PoseToPositionAndRot(`$\mathbf{p}_w$`);`
$\mathbf{p}_s = \mathbf{l}_s(\mathbf{q}_s)$; `// Eq. (3.1)`
$\dot{\mathbf{p}}_s = \mathbf{J}_s \dot{\mathbf{q}}_s$; `// Eq. (3.2)`
$\overline{\mathbf{p}}_{sb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^b\mathbf{R}_s \left(\begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_3 \end{bmatrix} \mathbf{p}_s - \overline{\mathbf{p}}_{sc} \right)$; `// Eq. (5.21)`
$\dot{\mathbf{F}} = (\mathbf{F} - \mathbf{F}_{prev})/T_s$; `// Derivative`
$\dot{\overline{\mathbf{p}}}_{sb} = (\overline{\mathbf{p}}_{sb} - \overline{\mathbf{p}}_{sb,prev})/T_s$; `// Derivative`
$\dot{\overline{\mathbf{p}}}_{sb,ref} = (\overline{\mathbf{p}}_{sb,ref} - \overline{\mathbf{p}}_{sb,ref,prev})/T_s$; `// Derivative`
$\ddot{\overline{\mathbf{p}}}_{sb,ref} = (\dot{\overline{\mathbf{p}}}_{sb,ref} - \dot{\overline{\mathbf{p}}}_{sb,ref,prev})/T_s$; `// Derivative`
$\sigma_{s1} = -1 + \left\lvert \frac{x_{sb}}{W} \right\rvert^m + \left(\frac{\max(\lvert y_{sb} \rvert - (H - W), 0)}{W} \right)^m$; `// Eq. (5.20)`
$\phi_{s1} = \sigma_{s1} + K_{s1} \left(\sigma_{s1} - \sigma_{s1,prev} \right)/T_s$; `// Eq. (5.22)`
$\boldsymbol{\phi}_{s2} = \begin{bmatrix} F_z - F_{z,ref} + K_{s2,z} \dot{F}_z \\ F_\alpha + K_{s2,\alpha} \dot{F}_\alpha \\ F_\beta + K_{s2,\beta} \dot{F}_\beta \end{bmatrix}$; `// Eq. (5.29)`
$\mathbf{A}_{s1} = \mathrm{pos}(\phi_{s1})\, K_{s1}\, \mathbf{H}_{s1}\, \mathbf{J}_s$; `// Eq. (5.25)`
$b_{s1} = -\mathrm{pos}(\phi_{s1})\, u_{s1}^+$; `// Eq. (5.25)`
$\mathbf{A}_{s2} = \mathbf{K}_{s2}\, \mathbf{H}_{s2}\, \mathbf{J}_{sn}$; `// Eq. (5.31)`
$\mathbf{b}_{s2} = -\overline{\mathbf{W}}_{s2}\, \mathrm{sign}(\boldsymbol{\phi}_{s2})\, u_{s2}^+$; `// Eq. (5.31)`
$\mathbf{A}_{s3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{R}_s & \mathbf{O}_3 \end{bmatrix} \mathbf{J}_s$; `// Eqs. (5.33) and (5.34)`
$\mathbf{b}_{s3} = \ddot{\overline{\mathbf{p}}}_{sb,ref} + K_{s3,v}\, \dot{\overline{\mathbf{e}}}_{sb} + K_{s3,p}\, \overline{\mathbf{e}}_{sb} + \mathrm{sign}\left(\dot{\overline{\mathbf{e}}}_{sb} + \frac{K_{s3,p}}{K_{s3,v}} \overline{\mathbf{e}}_{sb} \right) u_{s3}^+$; `// Eq. (5.33)`
$\ddot{\mathbf{q}}_{sc,1} = \mathbf{A}_{s1}^\dagger b_{s1}$; `// Eq. (3.5), i = 1`
$\mathbf{N}_{s1} = \mathbf{I} - \mathbf{A}_{s1}^\dagger \mathbf{A}_{s1}$; `// Eq. (3.6), i = 1`
$\ddot{\mathbf{q}}_{sc,2} = \ddot{\mathbf{q}}_{sc,1} + (\mathbf{A}_{s2}\mathbf{N}_{s1})^\dagger (\mathbf{b}_{s2} - \mathbf{A}_{s2}\ddot{\mathbf{q}}_{sc,1})$; `// Eq. (3.5), i = 2`
$\mathbf{N}_{s2} = \mathbf{N}_{s1}(\mathbf{I} - (\mathbf{A}_{s2}\mathbf{N}_{s1})^\dagger (\mathbf{A}_{s2}\mathbf{N}_{s1}))$; `// Eq. (3.6), i = 2`
$\ddot{\mathbf{q}}_{sc,3} = \ddot{\mathbf{q}}_{sc,2} + (\mathbf{A}_{s3}\mathbf{N}_{s2})^\dagger (\mathbf{b}_{s3} - \mathbf{A}_{s3}\ddot{\mathbf{q}}_{sc,2})$; `// Eq. (3.5), i = 3`
$\dot{\mathbf{q}}_{sc} = \ddot{\mathbf{q}}_{sc,3}\, T_s + \dot{\mathbf{q}}_{sc,prev}$; `// Integration`
$\mathbf{q}_{sc} = \dot{\mathbf{q}}_{sc}\, T_s + \mathbf{q}_{sc,prev}$; `// Integration`
`SendToSTRJointControllers(`$\mathbf{q}_{sc}$`);`
$\mathbf{q}_{sc,prev} = \mathbf{q}_{sc}$; `// For next iteration`
$\dot{\mathbf{q}}_{sc,prev} = \dot{\mathbf{q}}_{sc}$; `// For next iteration`
$\sigma_{s1,prev} = \sigma_{s1}$; `// For next iteration`
$\mathbf{F}_{prev} = \mathbf{F}$; `// For next iteration`
$\overline{\mathbf{p}}_{sb,prev} = \overline{\mathbf{p}}_{sb}$; `// For next iteration`
$\overline{\mathbf{p}}_{sb,ref,prev} = \overline{\mathbf{p}}_{sb,ref}$; `// For next iteration`
$\dot{\overline{\mathbf{p}}}_{sb,ref,prev} = \dot{\overline{\mathbf{p}}}_{sb,ref}$; `// For next iteration`

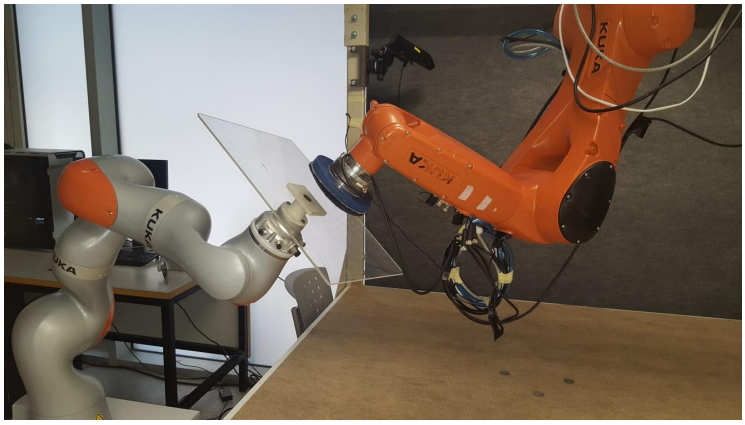


Figure 5.2: Experimental setup. STR: a 6R serial manipulator with an F/T sensor, a tool consisting of a cylinder (blue) and a piece of cloth attached to it (black). WR: a 7R cobot serial manipulator with a methacrylate flat workpiece attached to its end-effector by means of a self developed adaptor (white).

Table 5.3: Denavit-Hartenberg parameters for the STR ($d_{tool} = 0.06$)

Link i	θ_i(rad)	d_i(m)	a_i(m)	α_i(rad)
1	q_1	-0.4	0.025	$\pi/2$
2	q_2	0	0.455	0
3	$q_3 - \pi/2$	0	0.035	$\pi/2$
4	q_4	-0.42	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	$-0.08 - d_{tool}$	0	π

Table 5.4: Denavit-Hartenberg parameters for the WR ($d_{tool} = 0.06$)

Link i	θ_i(rad)	d_i(m)	a_i(m)	α_i(rad)
1	$q_1 + \pi$	0.36	0	$\pi/2$
2	$q_2 + \pi$	0	0	$\pi/2$
3	q_3	0.42	0	$\pi/2$
4	$q_4 + \pi$	0	0	$\pi/2$
5	q_5	0.4	0	$\pi/2$
6	$q_6 + \pi$	0	0	$\pi/2$
7	q_7	0.152	0	$\pi/2$

An external computer using *Ubuntu 16.04* ans O.S. and *ROS Kinetic* distribution has been used to implement the different algorithms specified in Section 5.3, and an XBox controller allows the user to teleoperate both robot arms simultaneously. The robot arms, F/T sensor and external computer were communicated by an Ethernet switch. It must be noted that, the KUKA LBR iiwa robot has two available Ethernet ports, of which, particularly, the one correspondent to the FRI (Fast Robot Interface) was used to communicate the robot with the external computer, as it allows sampling periods up to 2 ms. On the other hand, the remote controller is communicated with the computer via a USB connection.

5.4.2 Experiment conditions and parameter values

The values of the parameters of the two control algorithms were empirically tuned to obtain a proper performance of both robot arms.

The parameter values for the WR are as follows:

- WR control rate: $T_w^{-1} = 200$Hz.
- Parameters of WR Level 1: $K_{w1} = 1.0$, $u_{w1}^{+} = 4.0$, $W = H = M = 0.06$, $m = 4$ and the position and orientation of the superellipsoid match the initial value of the position and orientation of workpiece (which are

given by the WR end-effector pose), respectively, i.e., $\overline{\mathbf{p}}_{wc} = \overline{\mathbf{p}}_w(0)$ and ${}^b\mathbf{R}_w = {}^{wp}\mathbf{R}_w(0)$.

- Parameters of WR Level 2: $\mathbf{K}_{w2} = \text{v2dm}([0.5\ \ 0.5\ \ 2])$, $\mathbf{W}_{w2} = \mathbf{I}_3$, $u^+_{w2} = 1.0$, $\alpha_{w,max} = \beta_{w,max} = \gamma_{w,max} = 15°$ and the mid-range values for the workpiece orientation angles correspond to their initial values, i.e., $\alpha_{w,mid} = \alpha_w(0)$, $\beta_{w,mid} = \beta_w(0)$ and $\gamma_{w,mid} = \gamma_w(0)$.
- Parameters of WR Level 3: $K_{w3,v} = 1.5$, $K_{w3,p} = 1.0$ and $u^+_{w3} = 0.001$.
- Parameters of WR Level 4: $K_{w4,v} = 0.8$ and $K_{w4,p} = 0.5$.

The parameter values for the STR are as follows:

- STR control rate: $T_s^{-1} = 75\text{Hz}$.
- Parameters of STR Level 1: $K_{s1} = 0.6$, $u^+_{s1} = 6.0$, $W = 0.07$, $H = 0.12$ and $m = 4$.
- Parameters of STR Level 2: $\mathbf{K}_{s2} = \text{v2dm}([1.522])$, $\mathbf{W}_{s2} = \text{v2dm}([0.40.60.6])$, $u^+_{s2} = 1.0$ and $F_{z,ref} = 20\text{N}$
- Parameters of STR Level 3: $K_{s3,v} = 1.8$, $K_{s3,p} = 1.3$ and $u^+_{s3} = 0.01$.

Note that the difference between both control rates is not problematic, since the STR control algorithm needs information sent by the WR control algorithm, which is ensured, as the former is slower than the latter, but the WR control algorithm does not need information coming from the STR control algorithm.

5.4.3 Results

Seven experiments were conducted in order to study and verify the performance of the two control algorithms working separately and together, and of the different control laws of which they are composed. These experiments are arranged in three groups:

- Two experiments to analyze the behavior of the WR control algorithm.
- Four experiments to analyze the behavior of the STR control algorithm.

– One experiment to analyze the behavior of the whole bimanual application.

Note that, in the videos below for each experiment, the upper right corner shows a virtual image of the bimanual robot workspace with the following information to assist the user during the teleoperation: the *blue* dots represent the boundary of the allowed area on the workpiece for the STR tool (i.e., the 2D modified superellipse), which turn *green* when the corresponding boundary constraint becomes active; the *static yellow* dot corresponds to the initial position of the workpiece center; the *purple* dot is the current position of the workpiece center, i.e., of the WR end-effector; the *cyan* dot is the reference position commanded by the user to the workpiece center, i.e., to the WR end-effector; the *red* dot is the current position of the STR tool; and the *moving yellow* dot is the reference position commanded by the user to the STR tool. Note that some of these dots do not apply for some experiments. For instance, in the video of Experiment 1, which is focused only on the WR, the red and moving yellow dots are not shown since they are related to the STR.

5.4.3.1 Experiments for the WR control algorithm

There are two experiments belonging to this section: the first experiment (Experiment 1) focuses mainly on the study of the first control level, i.e., the boundary constraint, whereas the second experiment (Experiment 2) focuses mainly on the study of the second control level, i.e., the angular restriction. Moreover, both experiments study the behavior of the remaining control levels (third, reference tracking, and fourth, redundancy resolution), as in both cases the WR is teleoperated.

In Experiment 1 the WR is teleoperated (WR Level 3) so that the reference position commanded by the human operator is outside the allowed workspace, i.e., the 3D superellipsoid, thus activating the boundary constraint (WR Level 1) and keeping the workpiece inside the intended area, see the `video` (Video: Chapter 5, Experiment 1, 2021).

Several graphs are presented below in order to verify the quantitative performance of Experiment 1. In Fig. 5.3, the functions and activation of the boundary constraint in WR Level 1 are shown. Observe that this constraint is activated during two intervals, approximately from 19s to 50s and from 105s and 179s, and during the whole time of the experiment, the workpiece center is kept within the boundary, as can be seen in Fig. 5.4. Particularly, in Fig. 5.4 it

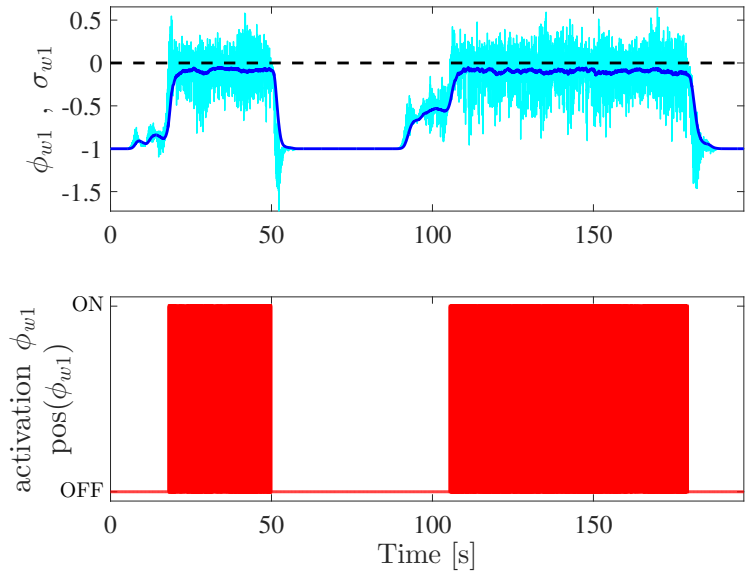


Figure 5.3: Experiment 1. WR Level 1: Top, constraint functions σ_{w1} (thick dark-blue) and ϕ_{w1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

can be appreciated that the reference position often surrounds the boundary, while the workpiece center travels along it without surpassing it. It also can be seen that this boundary is a superellipsoid relative to the initial position of the center of the workpiece.

In Experiment 2 the WR is teleoperated (WR Level 3) so that the angular references for the workpiece are beyond the allowed maximum angles (relative to its initial angular position), thus activating the angular restriction (WR Level 2) and keeping the workpiece angular position within the permitted interval, see the `video` (Video: Chapter 5, Experiment 2, 2021).

Fig. 5.5 shows the constraint functions of each one of the three restrictions. The restrictions are activated as follows to prevent the constraint functions $\sigma_{w2,i}$ surpassing the zero value: first, the pitch restriction is activated twice, then the roll restriction is also activated twice, then the yaw restriction is activated once and, finally, all restrictions are activated simultaneously. During the second activation of the roll restriction and the activation of the yaw restriction, the pitch angle reference remains close to its limit, so its restriction

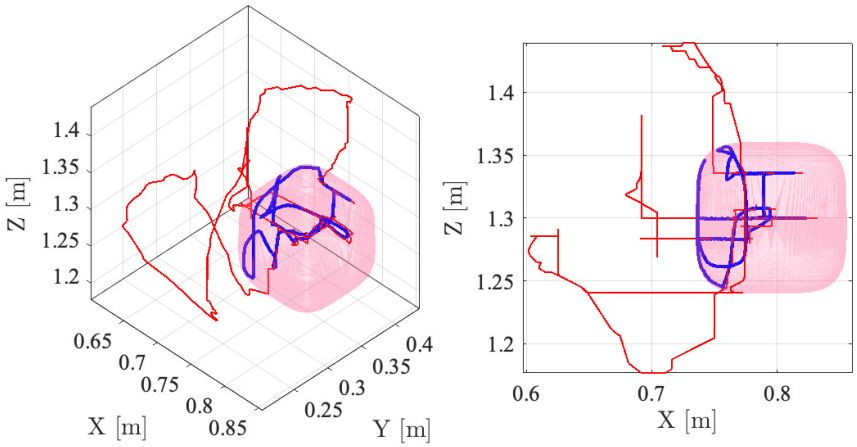


Figure 5.4: Experiment 1. 3D view (left) and top view (right) of the boundary constraint of WR Level 1: allowed region (pink mesh); actual position of the workpiece center (thick-blue line); and reference position for the workpiece center (thin-red line).

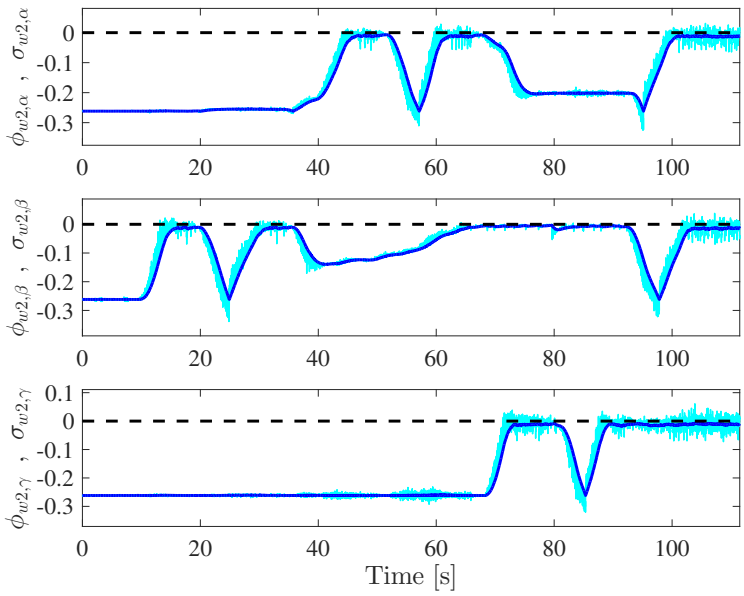


Figure 5.5: Experiment 2. WR Level 2: Constraint functions $\sigma_{w2,i}$ (thick dark-blue) and $\phi_{w2,i}$ (thin light-cyan) of the roll (α), pitch (β) and yaw (γ) angles of the workpiece and constraint limit (dashed).

is intermittently activated and deactivated. All this information is completed by observing Fig. 5.6, which shows how the angular reference goes beyond the permitted limits, while the actual position of the workpiece center always respects those limits.

5.4.3.2 Experiments for the STR control algorithm

As explained at the beginning of Section 5.4.3, there are four experiments belonging to this section: Experiment 3 and Experiment 4 study the behavior of the STR boundary constraint (STR Level 1), while Experiment 5 and Experiment 6 focus on the force and torque (F/T) control in order to keep contact and perpendicularity at all times, respectively (STR Level 2). Moreover, Experiment 3, Experiment 5 and Experiment 6 also include user teleoperation, so they also explore the behavior of the reference tracking control law (STR Level 3).

Experiment 3 shows the performance of the boundary constraint when it

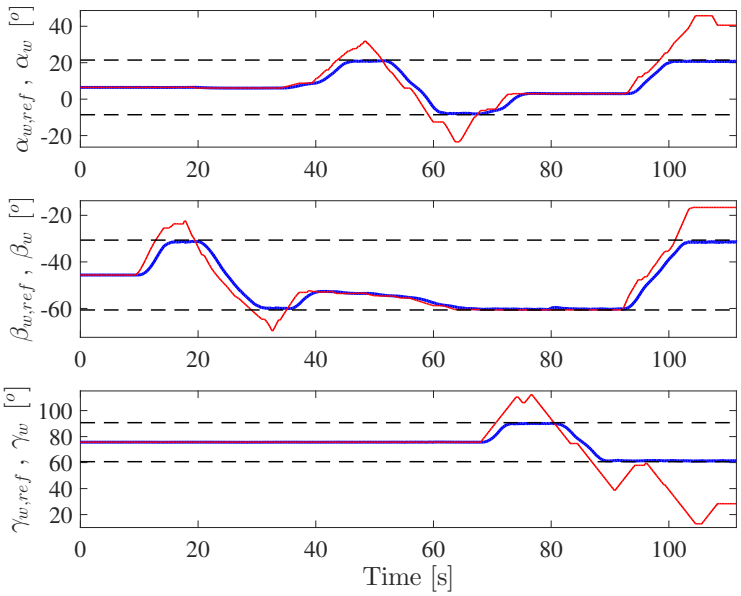


Figure 5.6: Experiment 2. Behavior of the restrictions of WR Level 2: angular reference (thin-red), actual angular position (thick-blue) of the workpiece and angular limits (dashed).

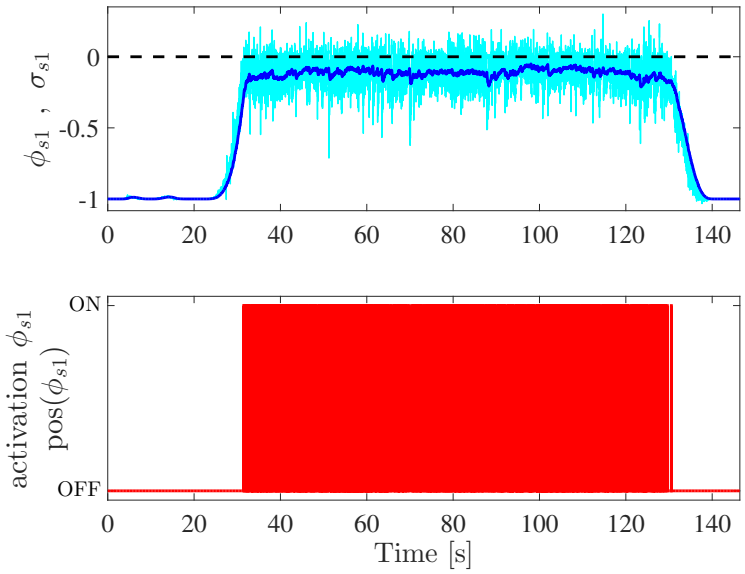


Figure 5.7: Experiment 3. STR Level 1: Top, constraint functions σ_{s1} (thick dark-blue) and ϕ_{s1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

remains static, in a similar fashion to Experiment 1 in Section 5.4.3.1. In this experiment, the STR is teleoperated (STR Level 3) so that the reference position for the tool tip in local coordinates of the workpiece frame tries to guide the tool beyond its permitted area of work, thus activating the boundary constraint and keeping the tool inside the allowed area, see the `video` (Video: Chapter 5, Experiment 3, 2021).

Several graphs are presented below in order to verify the quantitative performance of Experiment 3. In Fig. 5.7, the functions and activation of the boundary constraint in STR Level 1 are shown. Observe that this constraint is activated for an interval of approximately 131s (from 31s to 132s) and during the whole time of the experiment, the STR tool is kept within the boundary, as can be seen in Fig. 5.8. Thus, despite that the reference tries to take the tool tip out of the allowed area (i.e., the 2D modified superellipse), the actual tool tip always stays within due to the activation of the control law of the boundary constraint (STR Level 1), which has a higher priority than the reference tracking control law (STR Level 3).

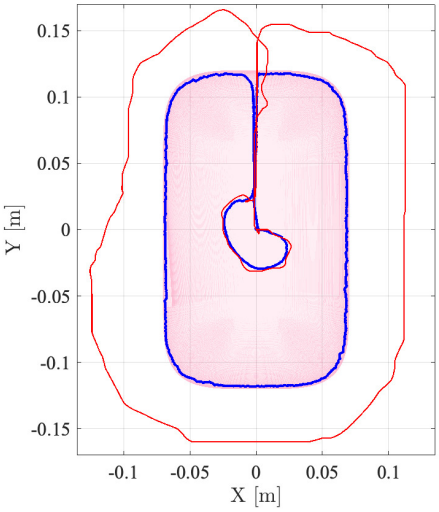


Figure 5.8: Experiment 3. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

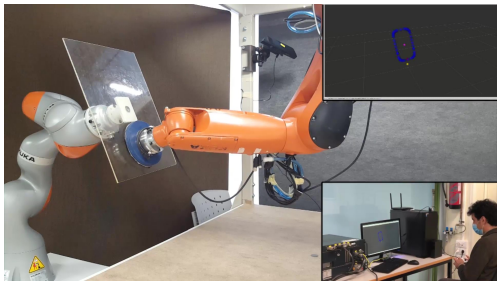
Several frames of the video of Experiment 3 depicted in Fig. 5.9 show how the surface treatment tool never leaves the workpiece area. This is due to the fact that the allowed area, a modified superellipse (3.9), has been calculated taking into account both the workpiece dimensions and the tool diameter, see Fig. 5.8.

Additionally, the quantitative performance of the surface treatment task can be seen in Fig. 5.10, where it is shown the behavior of the constraint functions $\{\sigma_{s2,z}, \sigma_{s2,\alpha}, \sigma_{s2,\beta}\}$, see (5.26)–(5.28). In particular, it can be seen that, thanks to the control law in Level 2 of the STR, all three functions are switching around zero, i.e., $\{F_z \approx F_{z,ref}, F_\alpha \approx 0, F_\beta \approx 0\}$, which means that the surface treatment is being performed properly. That is, the tool pressure and perpendicularity are kept regardless of the changes on the workpiece and treatment tool positions, which are commanded by the human user, and even though the sudden deformations of the methacrylate workpiece.

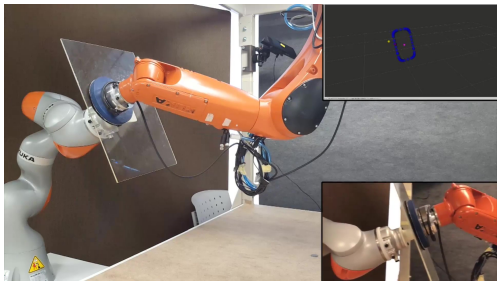
It should be noted that the STR control algorithm must take into account the movement of the workpiece, since the boundary is defined locally to the workpiece, so it will move during the bimanual task. Thus, another experiment (Experiment 4) in order to fully explore the first level of control is necessary, so as to study the behavior of the STR boundary constraint when it is the boundary itself that is moving and the surface treatment tool is trying to remain static.

Being so, in Experiment 4, the WR moves the workpiece along its end-effector X-axis (note that the end-effector Z-axis is along the last link of the robot, as usual) following a sinusoidal movement of amplitude 12cm and time period 30s, while the STR tries to keep the treatment tool still, so the boundary approaches the tool and its constraint goes active, pushing the tool and keeping it inside the permitted area, see the `video` (Video: Chapter 5, Experiment 4, 2021).

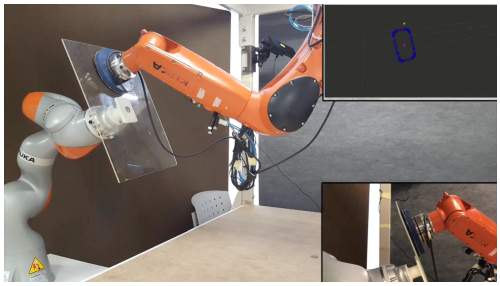
Observe that in Fig. 5.11, which presents a set of frames from the video of Experiment 4, the tool is pushed away from the workpiece limits when those limits reach the tool. In fact, the boundary constraint is activated before the tool reaches the limits, thanks to the variation in the distance between the tool and the boundary being computed in the modified constraint function ϕ_{s1} (see Section 3.2.3), allowing the STR to anticipate, as can be seen in graphs in Fig. 5.12, which show the functions and activation of the first control law, where ϕ_{s1} surpasses the threshold and activates the constraint well before σ_{s1} is in risk of surpassing it.



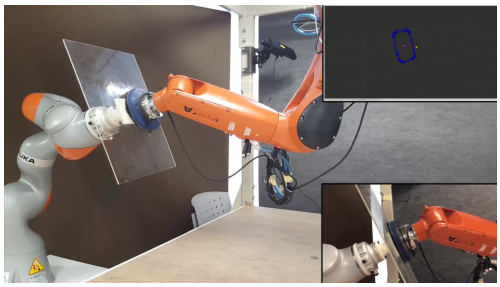
(a) video: 0m 40s



(b) video: 1m 08s



(c) video: 1m 31s



(d) video: 1m 55s

Figure 5.9: Frames of the video of Experiment 3.

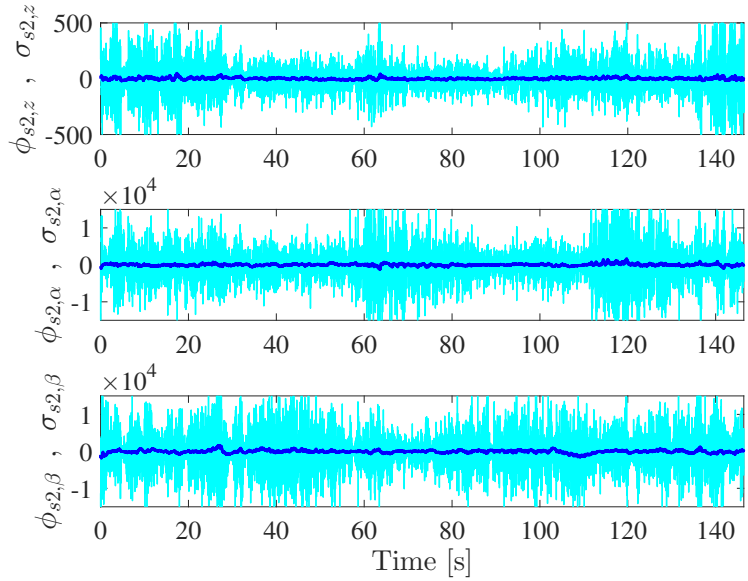
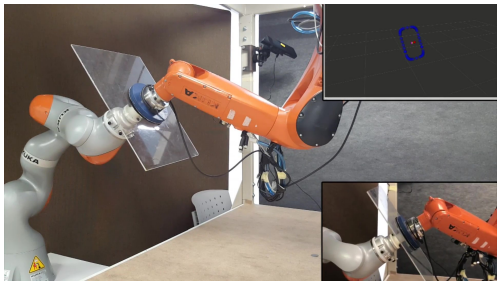


Figure 5.10: Experiment 3. STR Level 2: constraint functions $\sigma_{s2,i}$ (thick dark-blue) and $\phi_{s2,i}$ (light-cyan).

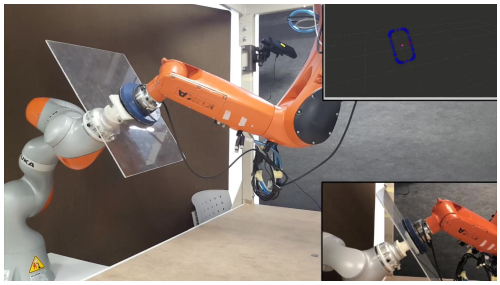
Note that, compared to the same functions in Fig. 5.7, in this case σ_{s1} is kept further away from the limit, since, in this case, the boundary is moving, so σ_{s1} changes faster, which is reflected in higher values of ϕ_{s1}. This anticipation is critical when the boundary is moving, as the STR has no control over how fast the boundary moves, so its reaction must be quick enough to avoid breaking the restriction. This is reflected in Fig. 5.13 as well, where the distance from the STR tool to the limits of the modified superellipse tends to be bigger than that displayed in Fig. 5.8 of Experiment 3, where the boundary remains static and, so, the surface treatment tool can approach the boundary more slowly.

Up next, a more detailed study of the performance of STR Level 2, i.e., force control in order to keep contact between the tool and the workpiece and torque control in order to keep perpendicularity between them, is necessary.

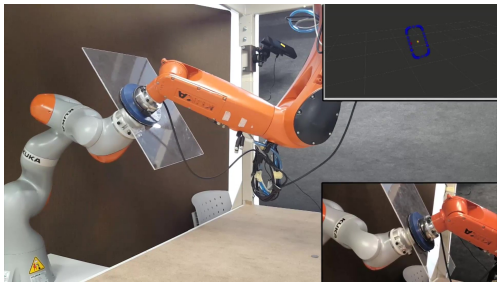
Experiment 5 poses a more challenging situation for STR Level 2 than the two previous ones. In this experiment, the WR moves the workpiece back and forth following its end-effector Z-axis with a sinusoidal movement of amplitude 5cm and time period 20s, thus pushing the surface treatment tool



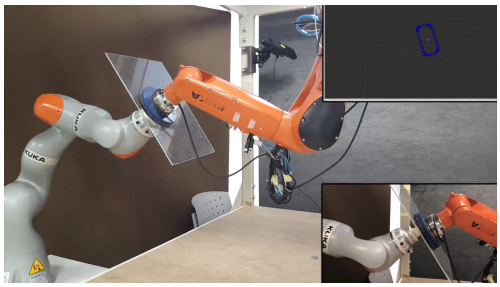
(a) video: 0m 11s



(b) video: 0m 15s



(c) video: 0m 26s



(d) video: 0m 32s

Figure 5.11: Frames of the video of Experiment 4.

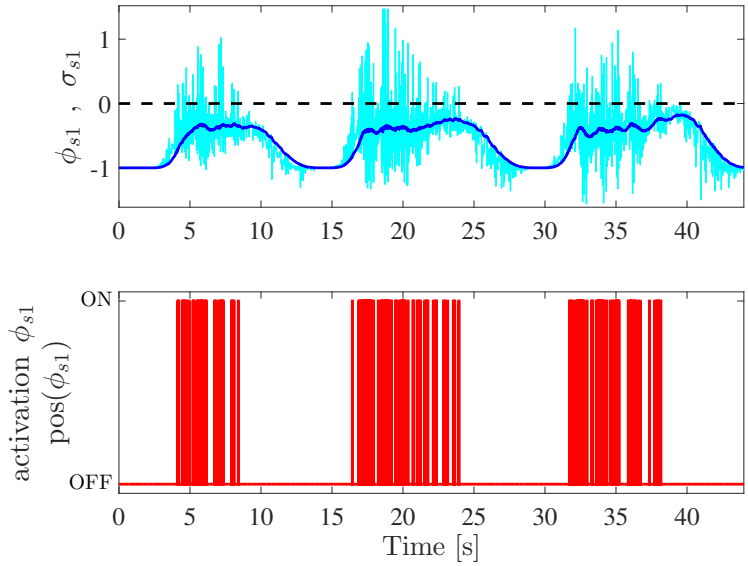


Figure 5.12: Experiment 4. STR Level 1: Top, constraint functions σ_{s1} (thick dark-blue) and ϕ_{s1} (thin light-cyan) and constraint limit (dashed); and bottom, constraint activation.

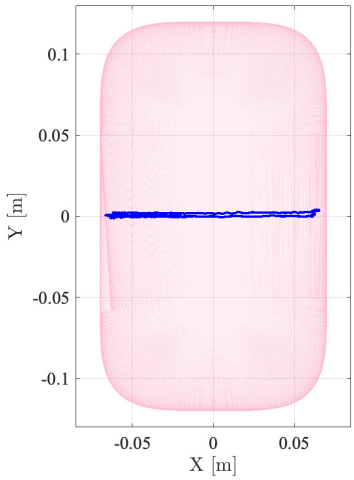


Figure 5.13: Experiment 4. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); and actual position of the STR tool (thick-blue line). Note that there is no reference position for the STR tool in this experiment, i.e., the STR tries to keep the treatment tool still.

while the STR is teleoperated while it tries to keep contact without impeding the workpiece advance. Eventually, the WR changes the course, forcing the STR control algorithm to adapt so as to not detach from the workpiece, see the `video` (Video: Chapter 5, Experiment 5, 2021). Note that during this experiment, the user is still able to teleoperate the STR and the boundary constraint works as expected, so the experiment offers a more complete outlook on the STR control algorithm and its capabilities.

Several frames of the video of Experiment 5 have been selected in Fig. 5.14 to show the moment of the change of course in the movement of the workpiece, and how contact is correctly maintained.

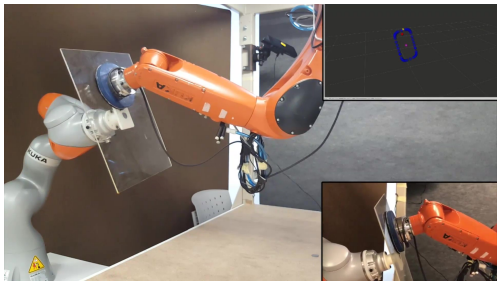
Graphs shown in Fig. 5.15 back this visual observation: during the whole experiment, pressure over the surface of the workpiece oscillates around the 20N reference, being its higher value 52.51 N and its lower value 2.33N, over 0N, so contact is ensured. Although it is not the main purpose of this experiment, these graphs also show how perpendicularity is kept thanks to the constant correction of the torque exerted over the F/T sensor, so it always oscillates about 0, without deviating to one side or the other.

Fig. 5.16 shows the roll and pitch angles of the surface treatment tool. Note that the orientation of the STR tool is kept relatively still, with variations between peaks of 9.25° and 5.72° in roll and pitch angles, respectively, which are small deviations taking into account that the workpiece is not a totally rigid surface, so the orientation must be adapted, especially when the tool approaches the limits of the workpiece, where it is less rigid.

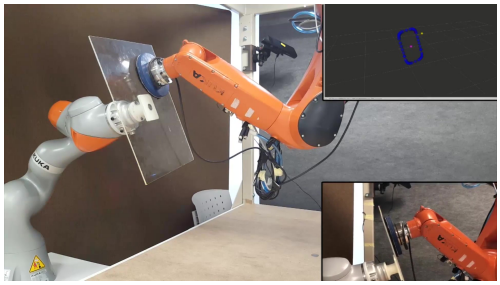
As explained above, the STR boundary restriction is also respected, as can be seen in Fig. 5.17

Finally, Experiment 6 in this section presents a situation similar to that of Experiment 5, but instead of moving the workpiece back and forth, the WR describes simultaneous angular movements, so the torque control behavior of STR Level 2 is explored more deeply. In this experiment, the workpiece describes an automatic sinusoidal movement of amplitude 15° for roll and pitch angles and 30° for yaw angle, and time period 40s for all three orientation angles. As it can be observed in the `video` (Video: Chapter 5, Experiment 6, 2021), the STR is able to adjust successfully to these changes while being teleoperated.

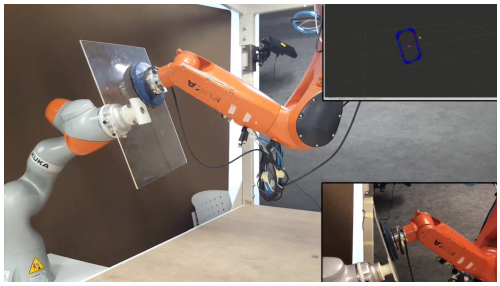
A set of frames from the video of Experiment 6 is presented in Fig. 5.18, showing different orientations of the workpiece. It must be noted that in all cases, the STR has been able to adapt correctly, keeping the tool orthogonal



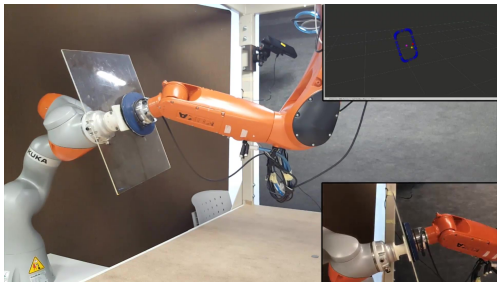
(a) video: 0m 44s



(b) video: 0m 57s



(c) video: 1m 00s



(d) video: 1m 09s

Figure 5.14: Frames of the video of Experiment 5.

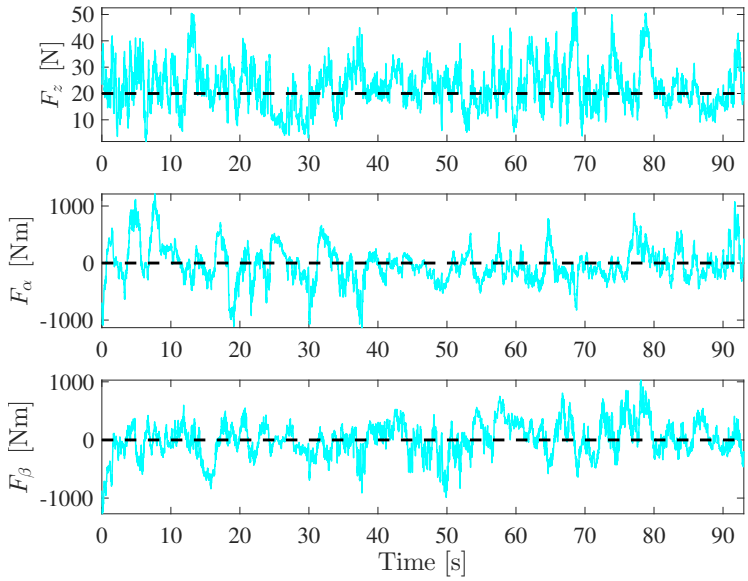


Figure 5.15: Experiment 5. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

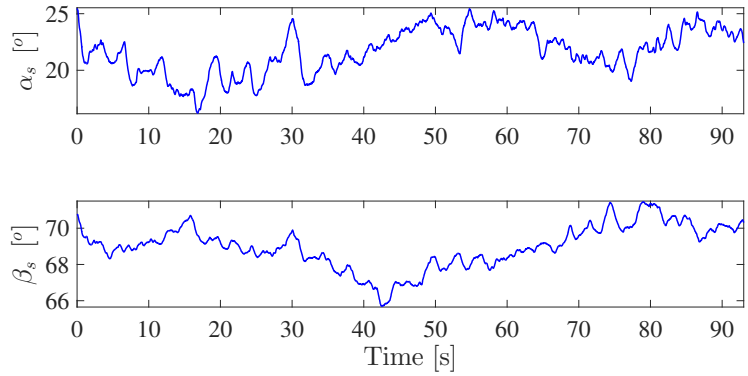


Figure 5.16: Experiment 5. Behavior of the restrictions of STR Level 2: roll (top) and pitch (bottom) angles of the STR tool.

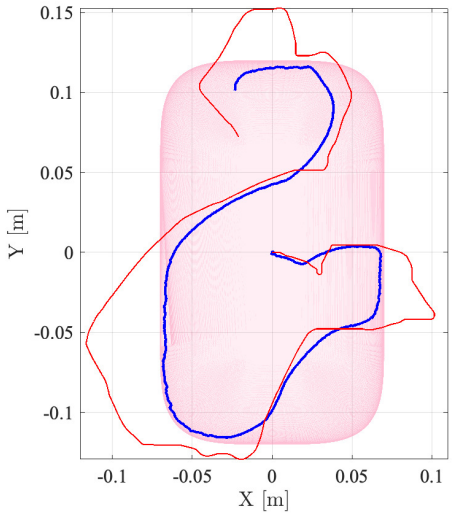


Figure 5.17: Experiment 5. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

to the surface of the workpiece without detaching from it.

Studying graphs in Fig. 5.19, which show F/T measurements, it is observed a similar situation to that of the former experiment, where perpendicularity is kept and that is reflected in the control law being able to keep the torque values oscillating around 0. Note that torque values do not stay in a negative or positive torque value for intervals longer than 3s (roll) and 4s (pitch) in the worst cases, which means the control law is able to adjust the orientation even when the workpiece orientation is changing and the STR is being teleoperated simultaneously.

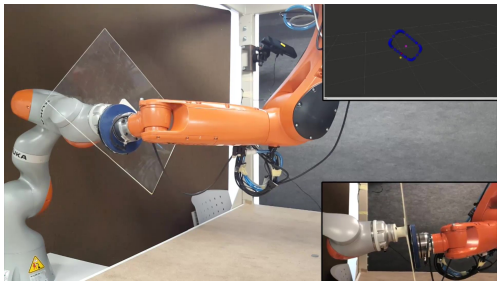
Moreover, similarly to Experiment 5 above, the STR boundary constraint is also activated and able to keep the surface treatment tool inside the allowed modified superellipse even though the human teleoperator tries to guide it beyond the boundary, as can be seen in Fig. 5.20.

5.4.3.3 Experiment for the bimanual application

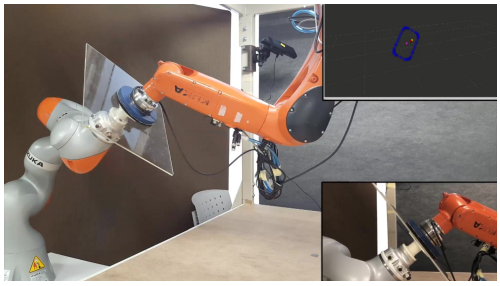
This section focuses on the practical study of this contribution as a whole, i.e., the bimanual assisted teleoperation for a surface treatment task. In this last experiment (Experiment 7) every factor studied in the last two sections is present: the user teleoperates the WR in global coordinates (i.e., 3D Cartesian position $[x\,y\,z]$ and orientation angles $[\alpha\,\beta\,\gamma]$) and the STR in local coordinates of the workpiece (i.e., 2D Cartesian position $[x\,y]$), and even though the task is complex, the different restrictions introduced by both WR and STR control algorithms and the coordination between them, allows the user to successfully carry out the task without detaching the tool from the workpiece, losing perpendicularity between them or leaving the intended workspace for both robot arms, see the `video` (Video: Chapter 5, Experiment 7, 2021). The surface treatment operation is represented by the brush of the surface treatment tool cleaning a translucent liquid spilled over the surface of the workpiece.

Although the fundamental quantitative aspects of the performance of the application have already been studied in Section 5.4.3.1 and Section 5.4.3.2, there are some specific challenges which arise from the interaction between the two robots and the simultaneous teleoperation of both robot arms by the human user.

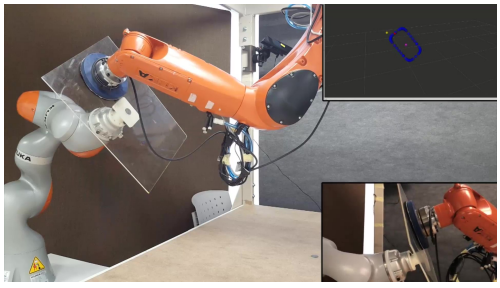
First, it must be noted that the angular movement limitation of the workpiece (WR Level 2) is still able to achieve its desired behavior even when the STR is exerting pressure in the most unfavorable area (that is, close to the



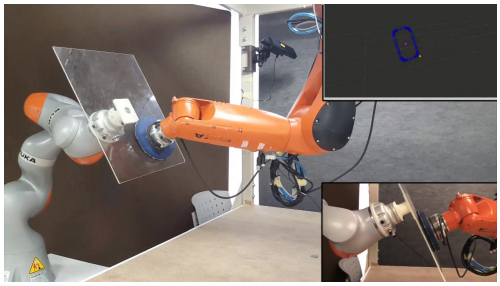
(a) video: 0m 24s



(b) video: 0m 44s



(c) video: 1m 10s



(d) video: 1m 53s

Figure 5.18: Frames of the video of Experiment 6.

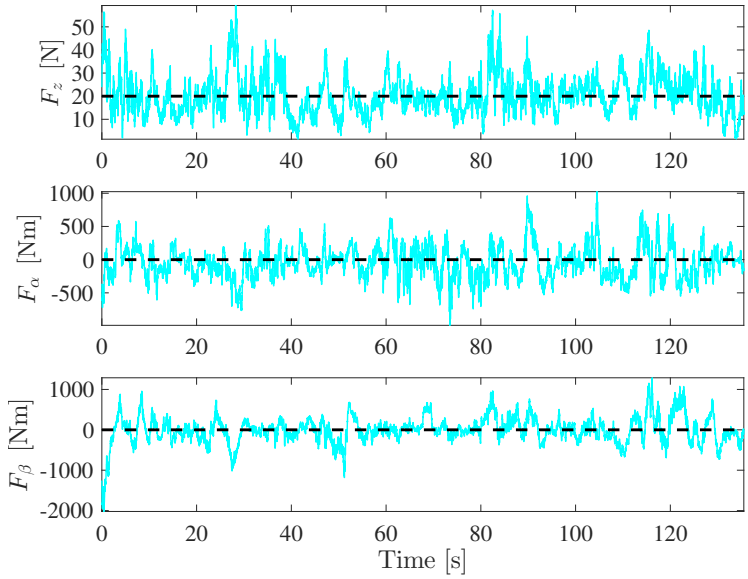


Figure 5.19: Experiment 6. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

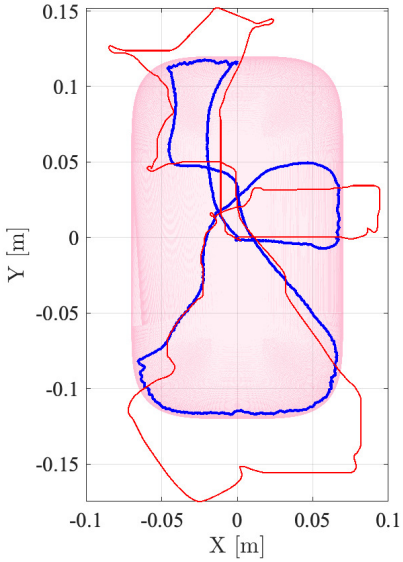


Figure 5.20: Experiment 6. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

workpiece limits on the side to which it is tilted). This situation is highlighted in the first and third frames in Fig. 5.21, and its quantitative performance can be observed in graphs depicted in Fig. 5.22, which show how, during the whole experiment, the angular positions of the workpiece are kept within the allowed limits.

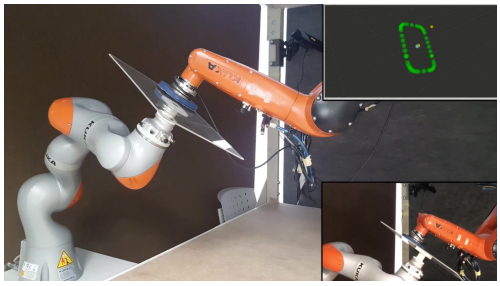
Second, it is also remarkable that the WR movements are now arbitrary, differently to what happened in Experiment 5 and Experiment 6 in Section 5.4.3.2, but the tool is still kept in contact with the workpiece and perpendicular to it, as it can be seen in the detail view (lower right corner) of the second and fourth frames in Fig. 5.21. The same as in Experiment 5 and Experiment 6, this performance can be checked by observing in Fig. 5.23 how force and torque keep oscillating around their reference points (20N and 0Nm, respectively), without significant continued deviations from them. This is thanks to the control law of STR Level 2 being always active.

And third, an operation which would be virtually impossible to complete by a direct control bimanual teleoperation is correctly carried out thanks to the action of the control algorithms. In order to fully observe this, the application as a whole must be studied, but additionally to what has already been shown, it can be noted that nor the workpiece neither the surface treatment tool abandon their allowed workspace even though their teleoperation reference positions try to command them to do so, see Fig. 5.24 and Fig. 5.25, due to the WR Level 1 and STR Level 1, respectively.

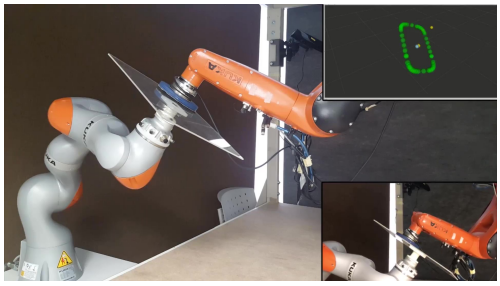
Finally, information about the control actions is presented in Fig. 5.26 and Fig. 5.27, corresponding to the WR and STR control algorithms, respectively.

In Fig. 5.26, note that WR Level 1 and Level 2 (boundary constraint and angular position restriction) only register activity when the teleoperator tries to break those restrictions (see Fig. 5.24 and Fig. 5.22), while WR Level 3 and Level 4 (reference tracking and redundancy resolution) area always active, although the reference tracking level is noticeably more active when it is trying to follow a reference that is blocked by WR Level 1 and Level 2, because the position error cannot be overcome.

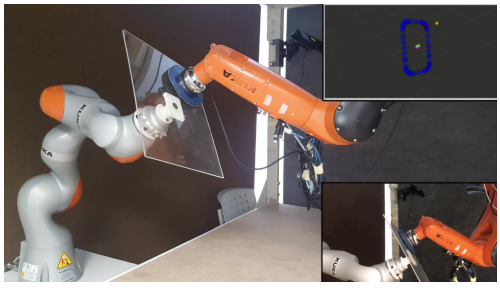
In Fig. 5.27, observe that STR Level 2 (F/T control) is always active because it is a conventional SMC, see Section 3.2.1 and Eq. (3.24), and STR Level 3 (reference tracking) is also active, since it is a continuous control. However, STR Level 1 only activates when the teleoperator tries to surpass the boundary, see Fig. 5.25.



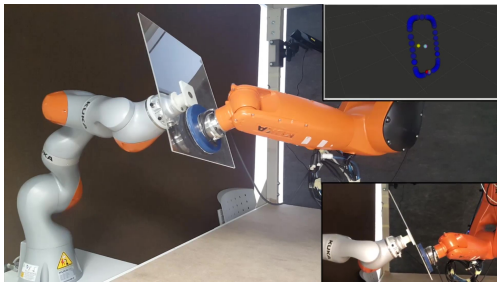
(a) video: 0m 53s



(b) video: 0m 55s



(c) video: 1m 03s



(d) video: 1m 52s

Figure 5.21: Frames of the video of Experiment 7.

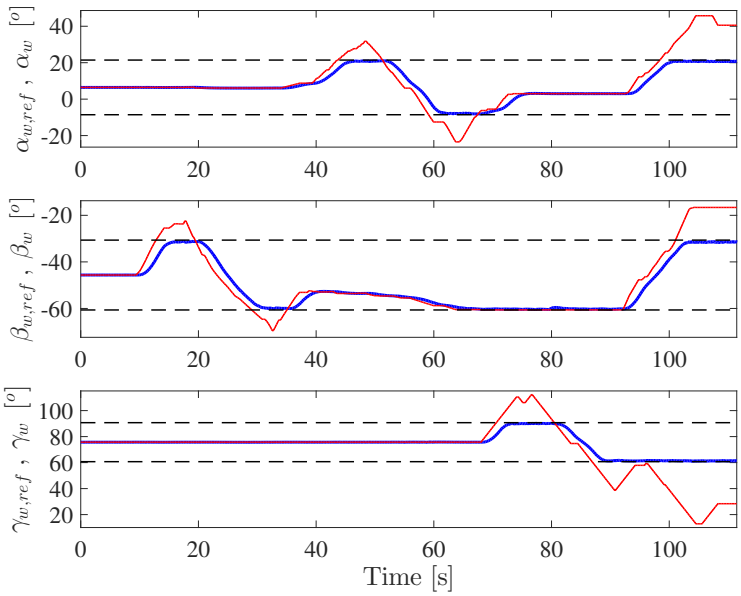


Figure 5.22: Experiment 7. Behavior of the restrictions of WR Level 2: angular reference (thin-red), actual angular position (thick-blue) of the workpiece and angular limits (dashed).

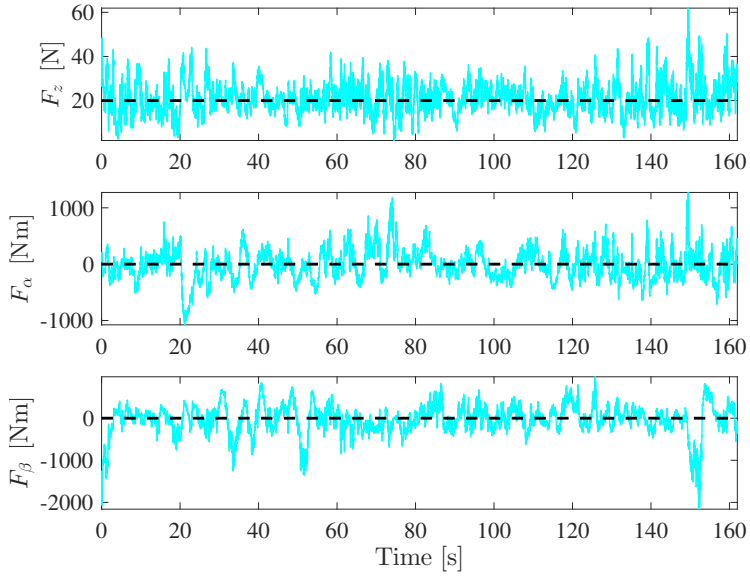


Figure 5.23: Experiment 7. Behavior of STR Level 2: measurements of the F/T sensor in the linear Z-axis (top), angular X-axis (middle) and angular Y-axis (bottom) of the STR end-effector frame. The reference value for each signal is represented with a dashed line.

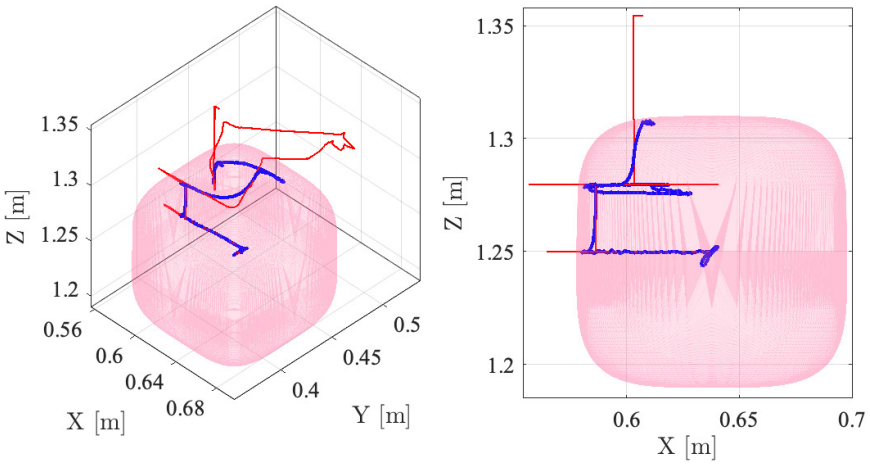


Figure 5.24: Experiment 7. 3D view (left) and top view (right) of the boundary constraint of WR Level 1: allowed region (pink mesh); actual position of the workpiece center (thick-blue line); and reference position for the workpiece center (thin-red line).

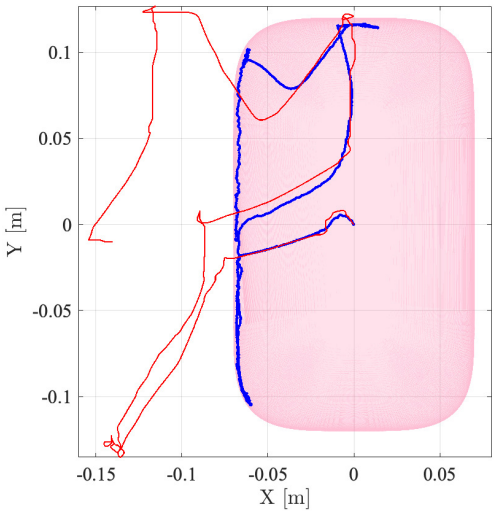


Figure 5.25: Experiment 7. Representation of the boundary constraint of STR Level 1: allowed region (pink mesh); actual position of the STR tool (thick-blue line); and reference position for the STR tool (thin-red line). Coordinates relative to the workpiece center.

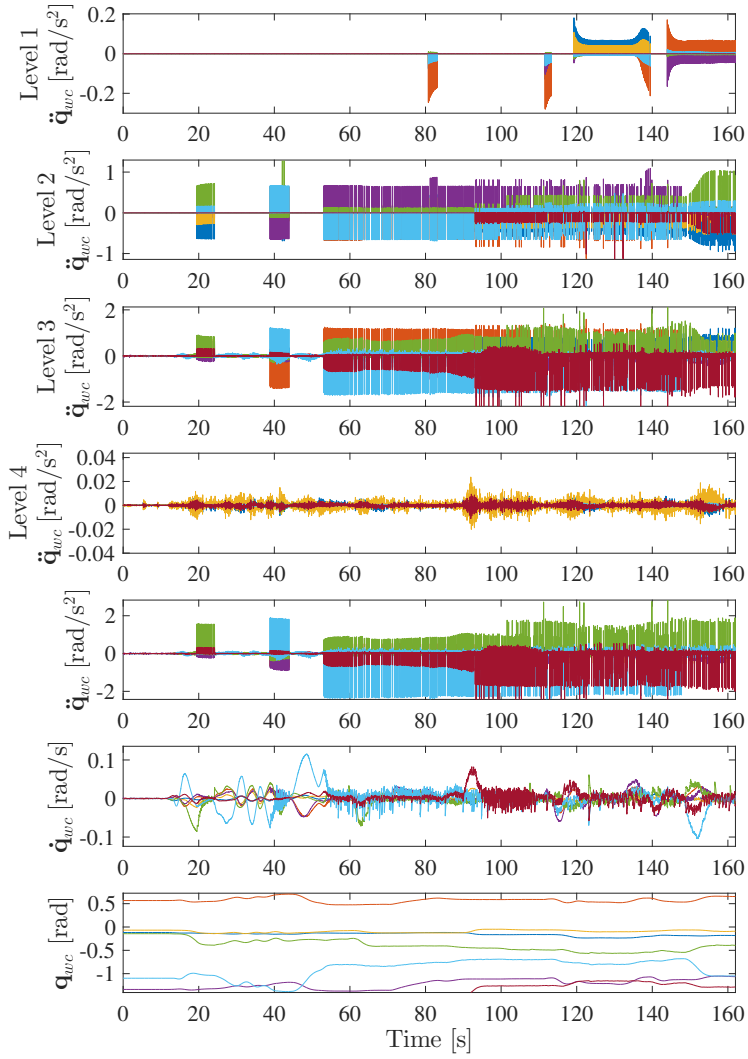


Figure 5.26: Experiment 7. Commanded joint actions for the WR: contribution of each priority level to the commanded joint accelerations in the first four plots, fifth plot represents commanded joint accelerations, sixth plot represents commanded joint velocities and seventh plot represents commanded joint positions.

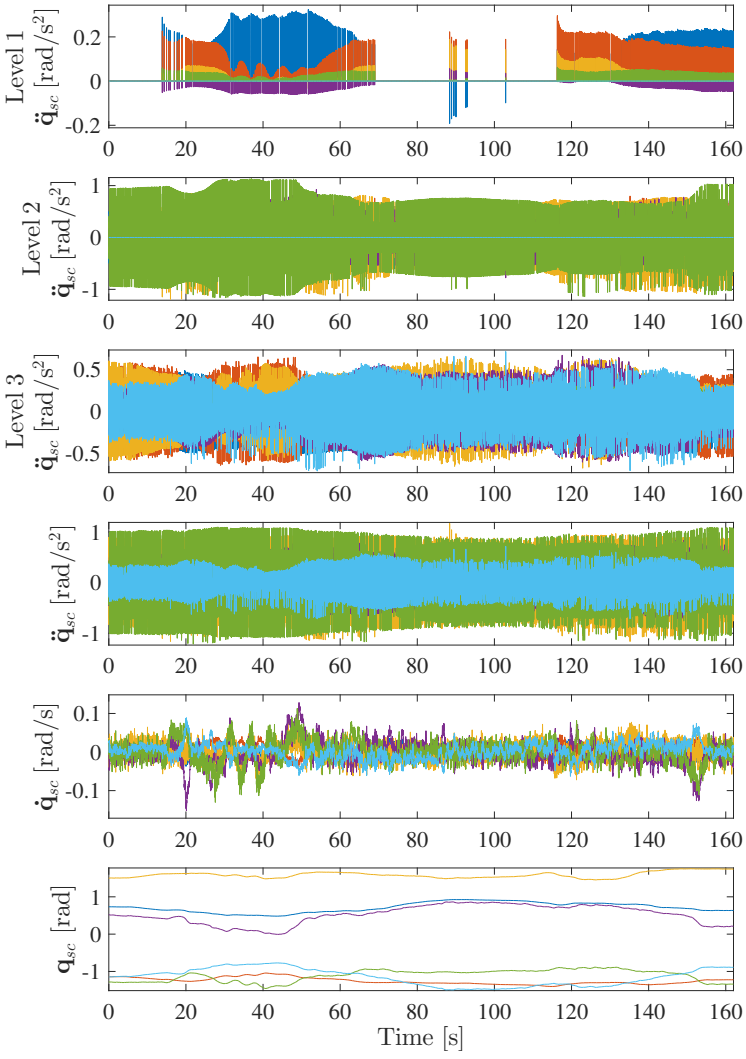


Figure 5.27: Experiment 7. Commanded joint actions for the STR: contribution of each priority level to the commanded joint accelerations in the first three plots, fourth plot represents commanded joint accelerations, fifth plot represents commanded joint velocities and seventh plot represents commanded joint positions.

5.5 Conclusions

A solution to perform surface treatment tasks has been developed in this work using a bimanual robotic system, i.e., two robot arms cooperatively performing the task. In particular, one robot arm holds the workpiece while the other robot arm has the treatment tool attached to its end-effector. In order to properly accomplish the surface treatment tasks, some robot coordinates were teleoperated by the human user, while the remaining robot coordinates were automatically controlled.

Furthermore, to assist the human user during the teleoperation, several constraints were defined for both robot arms in order to avoid exceeding the allowed workpsace. In particular, a boundary constraint was defined for each robot arm, as well as maximum orientation angles were considered for the robot arm that holds the workpiece.

A distinguishing feature of the bimanual robot control developed in this work is that not only conventional but also a one-side sliding mode control was used.

Furthermore, the feasibility and effectiveness of the method were shown through experimental results using two robot arms: a 6R industrial manipulator and a 7R cobot.

It is interesting to remark that, during the experimentation, the user found it difficult to teleoperate the robot arms, mainly because it was difficult for the user to figure out, in real-time, the spatial correspondence between the teleoperation commands and the robot movements. Therefore, in order to improve the user teleoperation, it is suggested as further work to develop an *advanced teleoperation* system based on mixed reality (e.g., an augmented reality headset) and haptic devices (e.g. a Phantom Omni), in order to make the robot teleoperation more intuitive to the user.

Chapter 6

Augmented reality-based interface for bimanual robot teleoperation

6.1 Introduction

6.1.1 Objective

Given the proliferation of applications based on Human-Robot Interaction, many approaches can be found in the literature proposing solutions for complex applications, where human and robot have to cooperate. However, many of these solutions do not take into account that human-robot interaction has to be natural and intuitive for the human (Cardoso, 2016; Kitson et al., 2017; Zhao and Allison, 2019; Solanes et al., 2020). Otherwise, the benefits that such cooperation may bring a priori will be negatively affected, and the cooperative solution adopted may be rejected.

Based on the complexity presented by the interaction of users with the so-called bimanual robotic systems, this work develops a novel methodology for the design of interfaces based on augmented reality so that this interaction is natural and intuitive for the user.

This chapter corresponds to a published contribution: (García et al., 2022b).

6.1.2 State of the Art

6.1.2.1 Bimanual robotics

The foundations and state of the art of bimanual robotics have already been discussed in Chapters 2 and 5. However, for the purpose of this Chapter, it is worth recalling that one of the reasons for the proliferation of *dual-arm robotic systems* is their behavior, quite similar to that of the human, which makes it possible for humans to relate their movements more intuitively (Smith et al., 2012; Makris et al., 2017; Sepúlveda et al., 2020).

Thus, the presence of the human interacting with the bimanual robotic system is very interesting due to the possibility of exploiting the human's natural knowledge of bimanual configurations and motions in order to improve the task performance (Ibarguren et al., 2020; Brantner and Khatib, 2021; García et al., 2022a).

In Chapter 5, an approach based on the task priority and sliding mode control techniques to perform surface treatment tasks using a bimanual robotic system is presented. In this case, the user was able to teleoperate all six Degrees-of-Freedom (DoF) of one robotic arm that holds the workpiece, the movement of which was limited in the 3D workspace, and to teleoperate two DoF of the other robotic arm, which holds the surface treatment, maintaining the right tool orientation and pressure.

6.1.2.2 Assisted robot teleoperation

As previously exposed in Chapters 2 and 5, teleoperation applications can be classified regarding the degree of shared control between the two extremes of direct control and supervisory control (i.e., almost fully automated robot behavior).

Telepresence (Niemeyer et al., 2016) allows the user to perform the robot teleoperation task by means of an interface, achieving a result less dependent on their skills.

For instance, the authors in (Lipton et al., 2018) proposed a low-cost telerobotic system based on virtual reality technology and the homunculus model of mind. In this case, the user was able to move both robotic arms according to the dynamic mapping between the user and the robot developed. In addition, the user was able to see the real workspace in the virtual environment using feedback from a camera. Similarly, the authors in (Bian et al., 2018) proposed

a virtual reality interface based on the three-dimensional coordinates of the shoulder, elbow, wrist and hand captured by a Kinect camera to model the geometry of the human arms and perform the mapping with the robot arms. As in (Lipton et al., 2018) the user receives visual feedback from a camera placed on the robot. In both cases, robot manipulation tasks were performed.

However, for more complex tasks (e.g., surface treatment tasks), interfaces developed with virtual reality techniques can increase the time of completion of the task and worsen the quality of the surface finishing, compared to that obtained by the human operator using direct teleoperation. This is due to two facts: on the one hand, when using virtual reality it is difficult to incorporate all the necessary information of the task in the virtual world and in real-time; and, on the other hand, the user already has a real notion of the robotic system and, hence, is able to guide it naturally and intuitively using direct teleoperation. For this reason, in order to get the best of both worlds (i.e., direct teleoperation and teleoperation based on virtual reality), the present work proposes to use interfaces based on augmented reality to provide a solution to a greater number of industrial tasks carried out with bimanual robots.

Other approaches try to make it easier the teleoperation of bimanual robotic systems such as in (Nicolis et al., 2018), where the authors developed a bimanual robot application in which a robot arm is teleoperated to grasp the workpiece, whilst the other robotic arm is automatically controlled using visual servoing in order to maintain the workpiece visible for the camera.

Since the performance of robot teleoperation may rely on the user skills, some approaches are focused on incorporating restrictions that prevent the user from commanding the robot to failure situations, such as Virtual Fixtures or the use of haptic devices, as discussed in 5.

Despite all the above, robot teleoperation by means interfaces and virtual barriers is still a subject of study due to the drawbacks it presents, mainly due to direct control performed by the user (Selvaggio et al., 2018). In this sense, this work presents a new methodology based on augmented reality devices to improve the current assisted teleoperation interfaces for bimanual robotics.

6.1.2.3 Augmented Reality-based Interfaces

Some previous works used AR interfaces to improve robot teleoperation for industrial tasks. For example, the authors in (Li et al., 2019) proposed a new AR interface to control a robot manipulator in order to facilitate the

interaction between the user and the robot. The authors in (Li et al., 2019) proposed a mixed reality system in order to move the end-effector of the robot system. The authors in (Rosen et al., 2019) proposed a mixed reality system to allow the user to visualize the intended teleoperation command previously to the real robot motion. A similar approach was developed in (Gadre et al., 2019), where a mixed reality head-mounted display enabled the user to create and edit robot motions using waypoints. Authors in (Mistry and Maes, 2009) proposed a multimodal AR interface coined as *Sixth Sense* that let the user interact with information that was projected onto physical objects through hand gestures, arm movements and, in some cases, blinking. Authors in Ismail et al. (2019) proposed a method for using hand gestures and speech inputs for AR multimodal interaction with industrial manipulators.

Note that most of the AR approaches mentioned above developed solutions for robot-object manipulation tasks. Thus, to the best of the author's knowledge, this is the first work that proposes a new AR interface for industrial complex tasks, such as surface treatment tasks involving a bimanual robot system.

In addition, the interaction with the robotic system needs to be natural and intuitive, not only from the point of view of the visual feedback produced by the AR but also from the point of view of the way of sending the robot commands. All AR headsets have interaction elements based on hand tracking. As demonstrated in (Cardoso, 2016; Kitson et al., 2017; Zhao and Allison, 2019; Solanes et al., 2020), such prolonged interaction over time can be annoying and not ergonomic enough. That is why, similarly to (Solanes et al., 2020), this work proposes the use of gamepads, which are devices ergonomically designed to be used for long periods of time.

As far as the author's knowledge, this is the first work proposing an AR interface together with a gamepad for bimanual robot teleoperation.

6.1.3 Proposed approach

This work develops an original augmented reality-based interface for teleoperating bimanual robots. The proposed interface is more natural to the user, which reduces the interface learning process. A full description of the proposed interface is detailed in the sections below, whereas its effectiveness is shown experimentally using two industrial robot manipulators. Moreover, the drawbacks and limitations of the classic teleoperation interface using joysticks

are analyzed in order to illustrate the benefits of the proposed augmented reality-based interface approach.

6.2 Previous work

Without loss of generality, this work uses the robotic application presented in 5 to demonstrate the benefits of the proposed AR-based interface with respect to conventional PC-based interfaces.

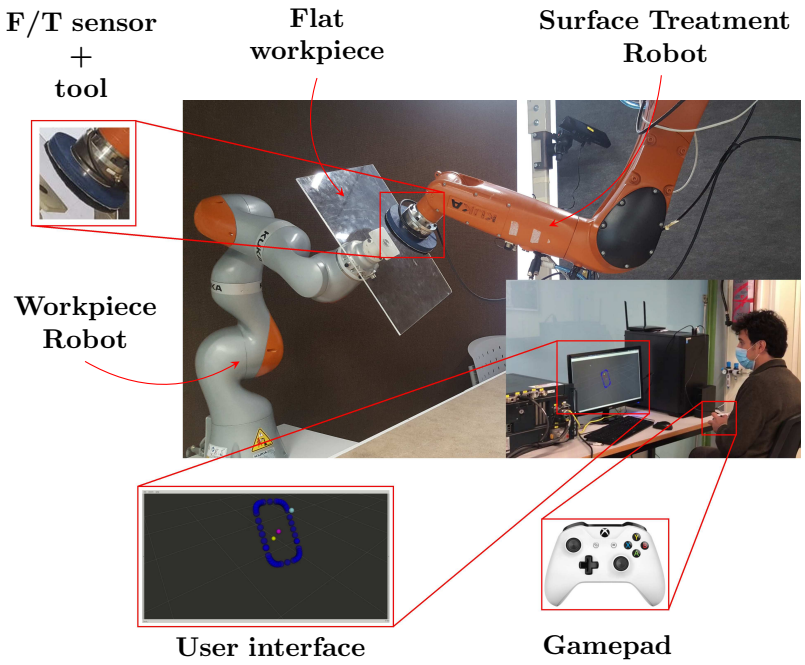
Since the application is already explained in detail in its correspondent chapter, this section focuses on the problems of using conventional PC-based interfaces, as the above mentioned application does.

The advanced bimanual robot teleoperation is based on the *task-priority strategy* (Nakamura et al., 1987; Nenchev, 1995) and *conventional* and *non-conventional Sliding Mode Controllers* (SMCs) (Garelli et al., 2011, 2007). As commented before, the goal of this bimanual robotic application is to perform a human-robot cooperative control loop so that the user operator partially teleoperates two robotic arms to perform a surface treatment operation, whilst the robots automatically keep the right tool force and orientation, see Fig. 6.1.

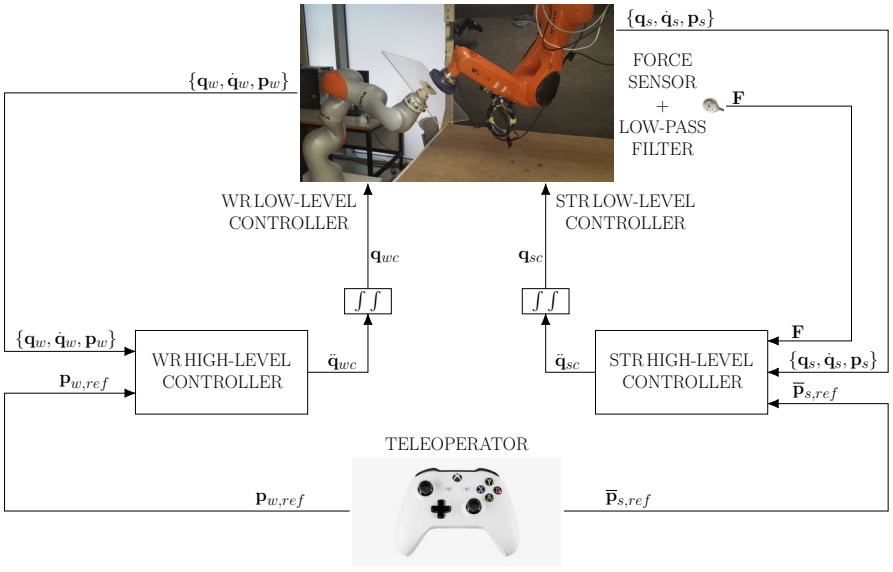
Thus, the so-called *workpiece robot* (WR), which consists of a 7R collaborative robot with a workpiece of flat methacrylate fixed to the end-effector using a self-made piece (see Fig. 6.1(a)), is in charge of holding the workpiece. Whereas, the so-called *surface treatment robot* (STR), which consists of a 6R robotic arm with a Force/Torque (F/T) sensor and a cylinder-shaped tool with a piece of cloth (see Fig. 6.1(a)), operates with the surface treatment tool on the workpiece. Thus, the user commands the workpiece position and orientation and, simultaneously, commands the 2D tool motion on the workpiece surface using an interface, which consists of a gamepad to command the robots and a visual feedback screen to show the user the robots and the user reference states, see Fig. 6.1(a).

6.2.1 Description of the conventional PC-based interface

The application in Chapter 5 presents a conventional PC-based interface, which shows a 3D interface on a screen, that is composed of the following visual elements (see Fig. 6.2):

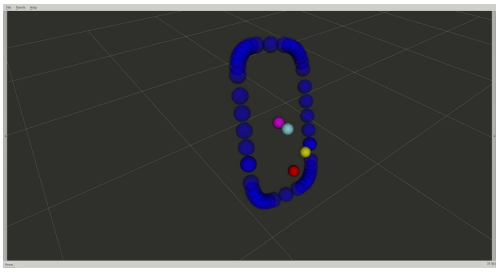


(a) Previous setup used for the real experimentation.

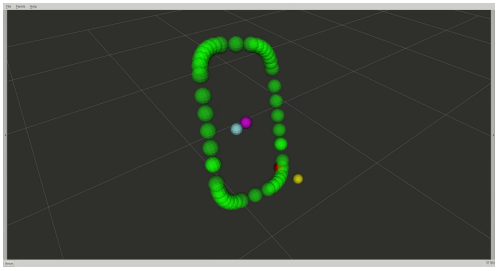


(b) Block control diagram for both robots (WR and STR).

Figure 6.1: Bimanual application setup and block diagram (for further details, refer to Chapter 5).



(a) Video: 0 m 20 s



(b) Video: 0 m 23 s

Figure 6.2: Conventional PC-based user interface: visual references and effects.

- STR reference consisting of a yellow sphere. The position of this element is commanded by the user using the gamepad input.

- STR current tool position consisting of a red sphere.

- STR boundary and WR workpiece orientation consisting of several blue spheres positioned along the curved define by the modified superellipse. When the tool collided with the boundary, the color of these spheres changes from blue to green.

- WR reference consisting of a cyan sphere. The position of this element is commanded by the user using the gamepad input.

- WR current workpiece position consisting of a pink sphere.

Note that the user commands both robots by means of the gamepad.

6.2.2 Discussion of human-robots interaction using conventional interfaces

The conventional PC-based user interface presents several problems that directly affect the task performance. Next, the three most relevant problems, which were identified from questions asked to several users that tested the application described above, are discussed.

The first significant problem reported by the users is that their interaction with the virtual environment was not natural. In particular, the robotic system is teleoperated in the 3D space and, hence, it requires changing the screen view to properly track the task. To do that, the user has to stop the robot teleoperation and accommodate the interface, affecting the total amount of time needed to complete the task.

The second significant problem reported by the users is that it was difficult for them to see the real system at any time, having to look to the real system instead of the interface. The reason given by several users, who exhibited this behavior, was that they needed to see what the real system was doing because they didn't know if the task was being done correctly or not. This means that this type of interfaces do not properly help the user to conduct the real task.

The third significant problem reported by several users is that it was difficult for them to move the references in the virtual 3D space, wasting a lot of time before resuming the robotic task.

All these issues and problems show the difficulties of using conventional interfaces and make evident the need to develop new interfaces allowing a more intuitive user interaction, especially when working with complex systems such as the bimanual robotic system considered in this work.

6.3 Proposed augmented reality-based user interface

In order to overcome the aforementioned problems of the conventional PC-based interface, this work proposes the use of AR technology to improve the user ergonomics and task performance. In particular, the conventional PC-based interface used in the previous setup (see Figure 6.1a) is replaced by an AR headset in the new setup—see Figure 6.3—allowing the user to see the relevant information in the form of holograms while still seeing at all times the real elements involved in the task: robots, workpiece, tool, etc.

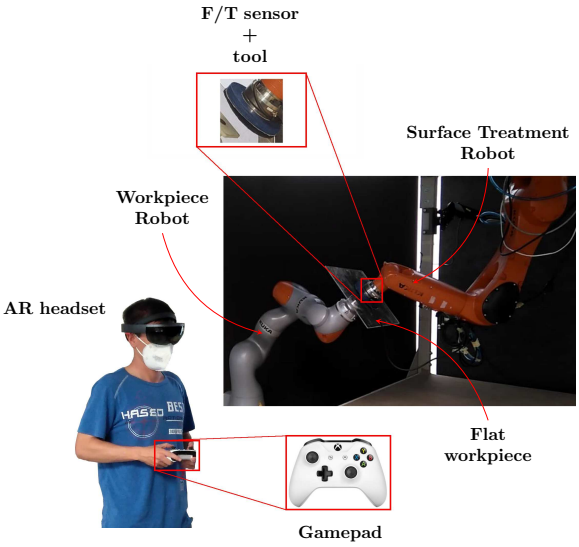


Figure 6.3: New setup used for the real experimentation.

Note that the remaining elements of the new setup (see Figure 6.3) are the same as in the previous setup (see Figure 6.1a): an STR with an F/T sensor and a cylinder-shaped tool with a piece of cloth; a WR with a flat workpiece of methacrylate attached to the end-effector using a self-made piece; and a gamepad to command both robots.

Figure 6.4 shows the methodology considered in this work to develop and validate the proposed AR-based interface. Although this methodology is used below to design the AR interface for the specific bimanual robot teleoperation task at hand, it is generic and, in general, it can be applied to design AR interfaces for other types of applications.

Firstly, the requirements of the applications were established based on the opinions of several users who previously tested the conventional PC-based interface. These requirements are summarized in Table 6.1.

A mockup design was developed taking into account this information. The designed AR-based interface has, from a functionality perspective, two kinds of virtual objects: firstly, those representing the STR and WR reference indicators; and, secondly, those corresponding to the boundaries information. In order to develop both kinds of virtual objects, several tools and strategies related to the mockup design were used. These preliminary designs were

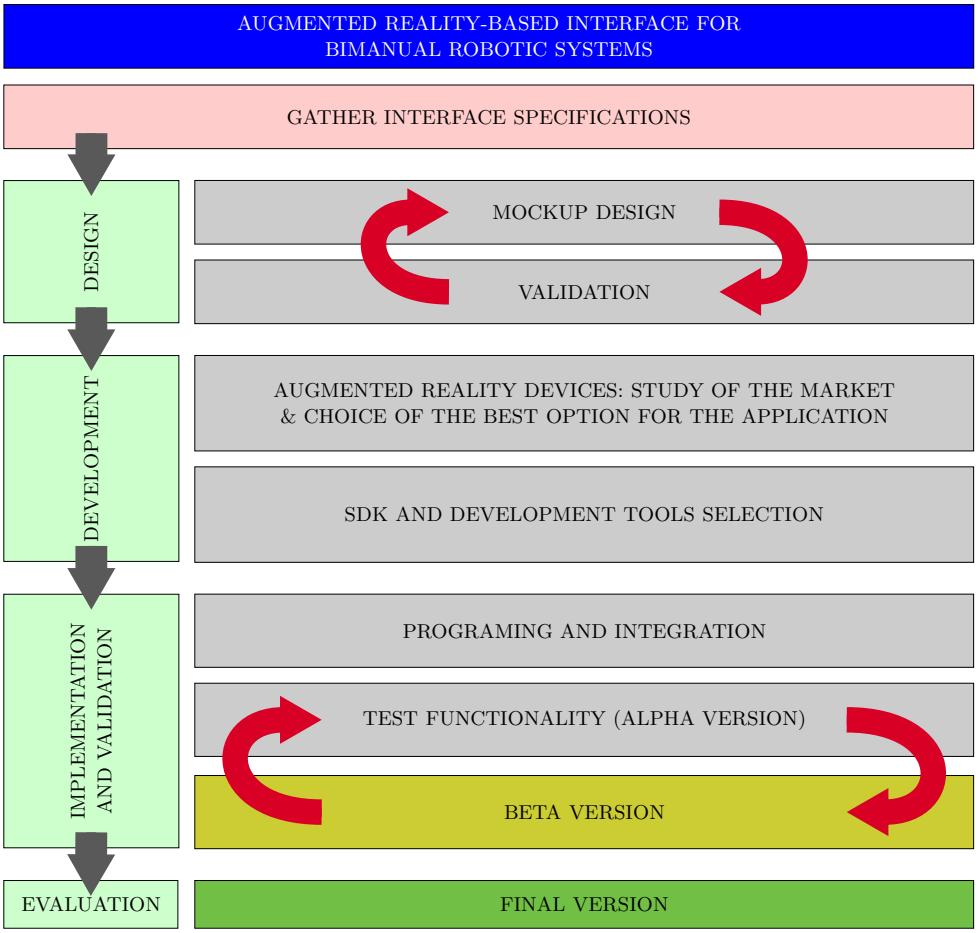


Figure 6.4: Flowchart of the methodology proposed in this work for designing the AR-based interface.

Table 6.1: Application requirements.

The user should have the option to see the full boundaries when required
The part of the boundary activated should be indicated (e.g., visually, sound, etc.)
STR tool reference direction should be indicated
WR rotation reference direction should be indicated
The new interface should use a similar interaction device to that of the previous PC-based interface (i.e., gamepad, joystick, or similar)
Alarm sounds should be used to indicate boundary activation
The user should have the option to remove all holograms
Holograms should not disturb the user visibility during the task
The user should have the option to configure, activate, and deactivate the alarm sounds

validated by some users before their implementation.

Once the preliminary design was finished, the following step was to study the best option of AR headset to be used for the application at hand. Several considerations were taken into account, especially the following: first, the capability of the device to be used in industrial environments; second, the stability of the holograms, which is important when working in this kind of application; third, the computational power of the device; fourth, the sound capabilities; and fifth, the communication capability (i.e., Bluetooth and WiFi). Note that most AR headsets in the market accomplish the aforementioned requirements. However, among all of them, Microsoft$^{\circ}$ *HoloLens glasses* (Microsoft Hololens, 2022) were chosen because the second generation of this device offers several services that could be added to the final version of the interface according to the company needs (Microsoft Hololens (2nd gen) hardware details, 2020).

Once the AR headset was selected, the interface was developed. Using a PC workstation, the proposed virtual objects were created and assembled in a virtual space using Blender 2.7 (Hess, 2010) and *Unity* (Jackson, 2015), respectively. This was an iterative design process, where the main characteristics of the virtual objects (e.g., size, color, shape, etc.) and their interactions were verified and modified, connecting the workstation with the AR headset in a remote mode from the Unity editor (note that the perception of the holograms is different when showing them in a PC screen compared to when projecting them in the real world through the AR headset), until the result

was satisfactory.

Figure 6.5 shows the holograms designed for the robot references. In the case of the WR, the user can command the robot through the 3D workspace and modify the end-effector orientation. For this reason, two different holograms were designed. The translation reference hologram was modeled by a 3D orange cube; see Figure 6.5a. This hologram appears when the user teleoperates the WR translation reference. To reduce the number of holograms present at any moment, this hologram disappears 3 s after the user has stopped moving the WR translation reference. The orientation reference hologram was modeled by an animated arrowed yellow circle; see Figure 6.5b. This hologram appears when the user teleoperates the WR rotation reference, and disappears 3 s after the user has stopped moving the WR rotation reference. It should be noted that, in both cases, the movement of the references is relative to the position of the user, i.e., the AR headset, making their use more intuitive and natural. The STR translation reference was modeled by a yellow arrow attached to a green sphere; see Figure 6.5b. Note that this hologram is constrained to the plane of the workpiece surface, allowing a 2D movement. This hologram disappears 3 s after the user has stopped moving the STR translation reference.

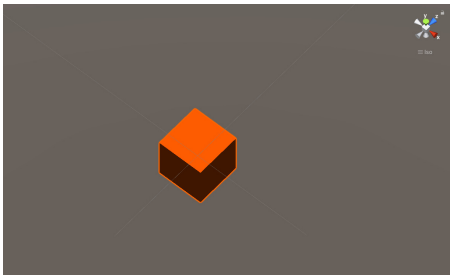
Figure 6.6 shows the holograms designed for the 2D and 3D boundaries.

The 3D boundary is modeled by a superellipsoid—see Figure 6.6a—as defined in 3.1.3.

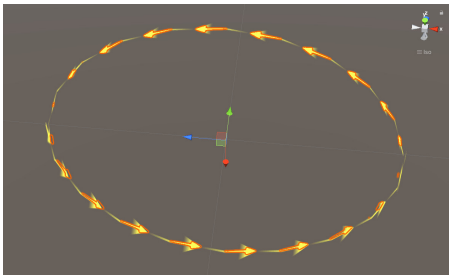
The 2D boundary is modeled by a modified superellipse—see Figure 6.6c—also as it is defined in 3.1.3.

Note that if the proposed boundary holograms were permanently shown, they could occlude some real elements from the user's view, affecting the task performance. For this reason, a new material shader (Unity, 2022) was designed; see Figure 6.7. This shader computes the minimum distance between the robot end-effector and the 3D boundary, for the case of the WR, or the closest point of the robot tool to the 2D boundary, for the case of the STR. Thus, the shader only displays the affected part of the boundary hologram. That is, as the WR end-effector and/or the STR tool approach to the 3D and 2D boundaries, respectively, the part of the boundary hologram affected is progressively displayed; see Figure 6.6b,d.

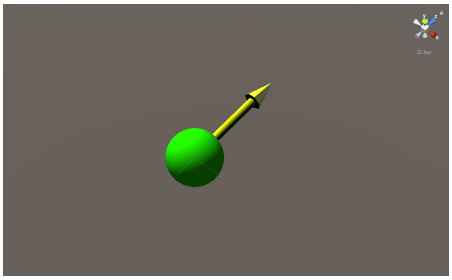
In addition to this, and according to the user requirements, two warning sounds were included in the interface: the first one to indicate that the STR tool is close to the 2D boundary; and the second one to indicate that the



(a) WR: translation reference hologram.

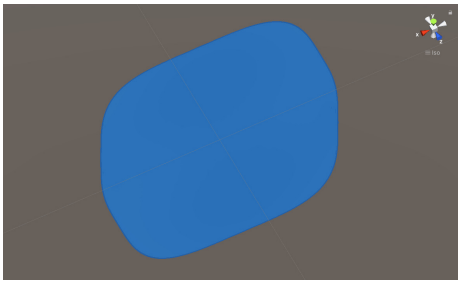


(b) WR: rotation reference hologram.

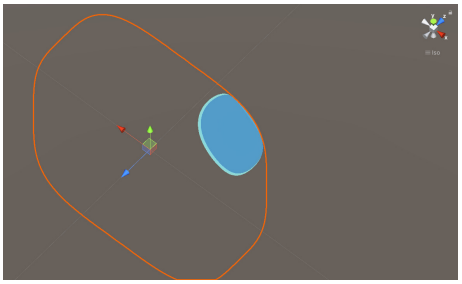


(c) STR: translation reference hologram.

Figure 6.5: Proposed holograms for the robot references.

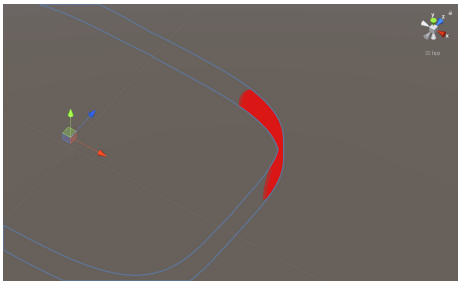


(a) 3D boundary hologram (full).



(b) 3D boundary hologram (local).

(c) 2D boundary hologram (full).



(d) 2D boundary hologram (local).

Figure 6.6: Proposed holograms for the robot 3D and 2D boundaries.

WR end-effector is close to the 3D boundary. Moreover, the user is able to deactivate this warning sound at any time.

Once the main holograms and sound elements were implemented, some communication protocols were used and programmed. Bluetooth communication between the Microsoft$^{\text{ő}}$ *HoloLens glasses* and the gamepad was established to allow the user to provide commands to the interface. Moreover, in order to avoid non-desired interactions with the interface, voice and gesture commands were deactivated by default. In addition, the AR interface and the robot controller communicate via WiFi with Protocol TCP/UDP at 10 Hz.

6.4 Results

This section presents four experiments to show the main functionalities of the developed AR-based interface; the performance of the 2D boundary and the STR reference hologram; the performance of the 3D boundary and the WR reference hologram; and the performance of the overall system when the user

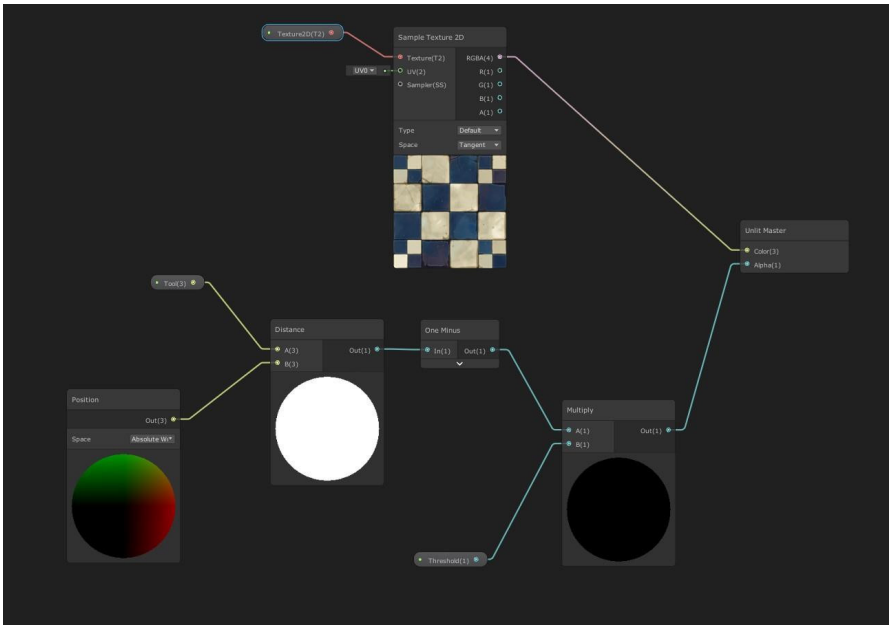


Figure 6.7: Material shader designed for controlling the visibility of the 3D and 2D boundaries depending on the proximity of the WR end-effector and STR tool, respectively.

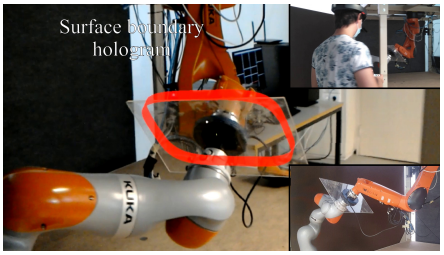
commands simultaneously both robots using the proposed AR-based interface.

Figure 6.8 depicts several frames of the *first experiment*, which shows the main functionalities of the AR interface implemented in the Microsoft© *HoloLens glasses*; see the video at (Video: Chapter 6, Experiment 1, 2022). Figure 6.8a shows the full 3D boundary hologram, whilst Figure 6.8b shows the full 2D boundary hologram. Note that both holograms are hidden by default. Figure 6.8c shows the WR end-effector translation reference hologram, whilst Figure 6.8e,f show the WR end-effector rotation reference hologram. Note that, in the case of the rotation, the animated arrows indicate the direction of the commanded angle while the yellow circle indicates the rotation in the roll, pitch, and yaw angles, or a combination of them. Figure 6.8d shows the STR reference hologram.

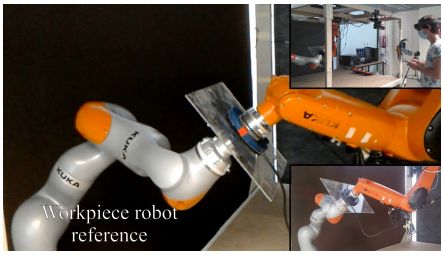
Figure 6.9 depicts several frames of the *second experiment*, which shows the performance of the 2D boundary and the STR reference hologram; see the video at (Video: Chapter 6, Experiment 2, 2022). Figure 6.9a shows how the user is commanding the STR tool towards one side of the workpiece and, when the tool approaches the 2D boundary, the boundary region closest to the STR tool is shown in red and the warning sound is activated; see Figure 6.9b,c. Note that, when the user reference exceeds the 2D boundary, the tool is automatically kept within the allowed region. More details about this aspect can be further analyzed in Figure 6.10, which shows the allowed region on the workpiece surface, the trajectory followed by the user reference, and the trajectory followed by the STR tool. Figure 6.9d shows how the 2D boundary hologram automatically disappears when the STR tool is far from the 2D boundary.

Figure 6.11 shows the position followed by the STR tool on the workpiece surface, which is due to the STR teleoperation, together with the reference values provided by the user. In particular, it can be appreciated that the trajectory described by the STR tool corresponds closely to the user reference values, except obviously when the 2D boundary constraint is active; see the bottom graph in Figure 6.11. In fact, the maximum deviation of the actual STR position values compared to the user reference values, when the 2D boundary constraint was not active, was around 3.2 cm, with a standard deviation of around 0.8 cm; see Table 6.2. Note that these teleoperation error values include all the potential sources of error: communication delays, high-level and low-level robot control, the accuracy of the workpiece location, teleoperation system, etc. Therefore, it can be concluded that the accuracy

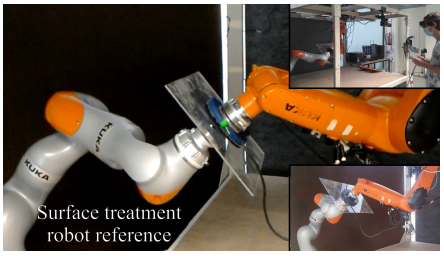
(a) Video: 0 m 20 s.



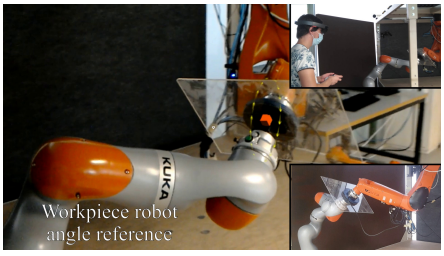
(b) Video: 0 m 23 s.



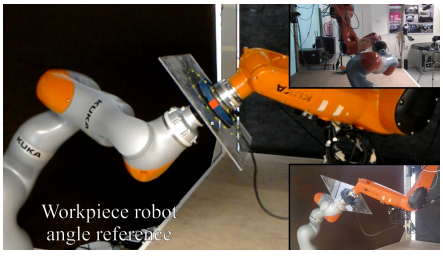
(c) Video: 0 m 30 s.



(d) Video: 0 m 36 s.

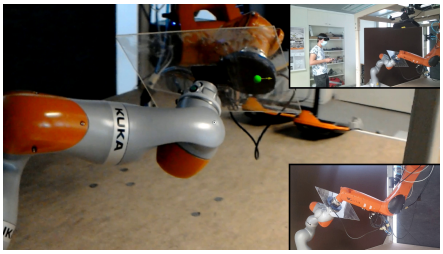


(e) Video: 0 m 43 s.

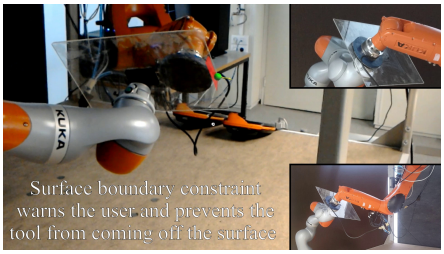


(f) Video: 0 m 57 s.

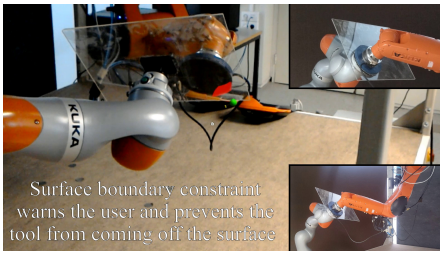
Figure 6.8: First experiment: frames of the video showing the functionalities of the proposed AR-based interface. See the video at (Video: Chapter 6, Experiment 1, 2022).



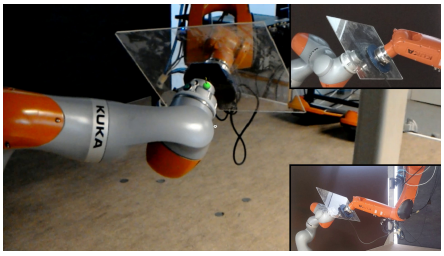
(a) Video: 0 m 31 s.



(b) Video: 0 m 41 s.



(c) Video: 1 m 05 s.



(d) Video: 1 m 46 s.

Figure 6.9: Second experiment: frames of the video showing the performance of the 2D boundary and the STR reference hologram. See the video at (Video: Chapter 6, Experiment 2, 2022).

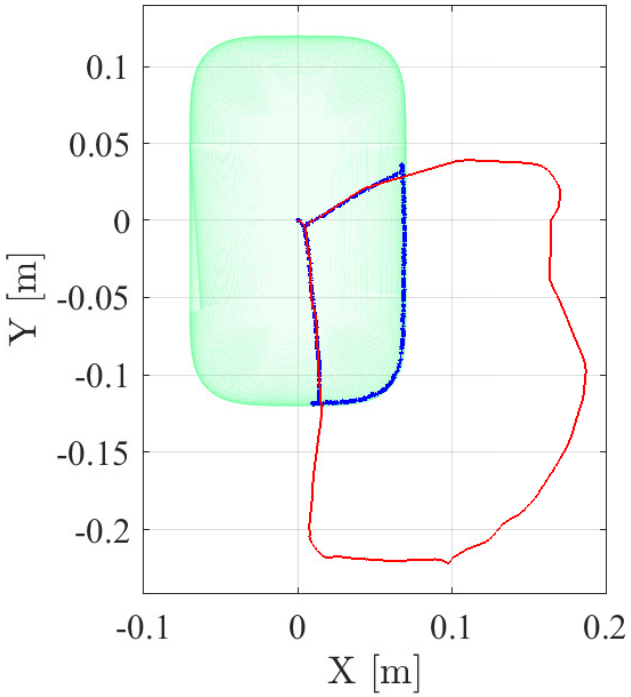


Figure 6.10: The 2D trajectory performance for the second experiment, showing the 2D boundary and the STR reference hologram (see the video at (Video: Chapter 6, Experiment 2, 2022)): 2D allowed workpiece region in green; trajectory followed by the user reference in thin red line; and trajectory followed by the STR tool in thick blue line.

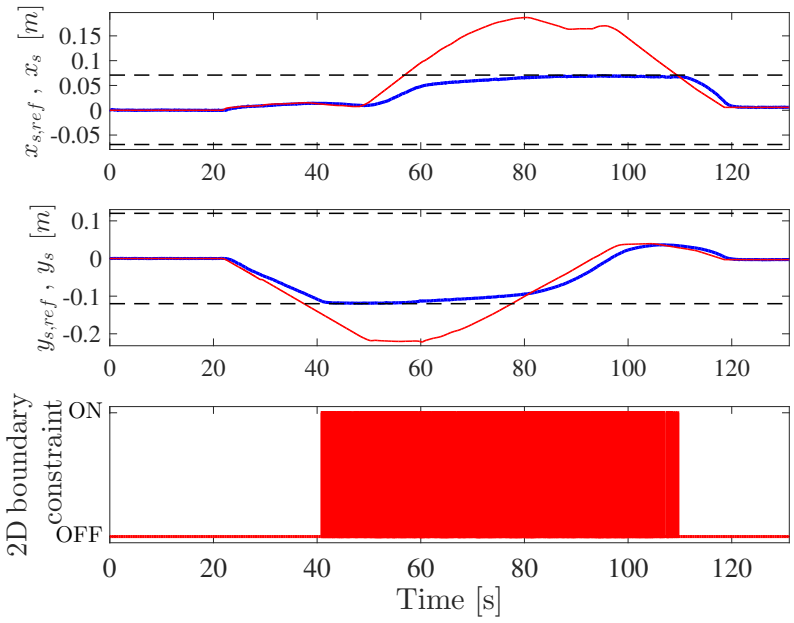


Figure 6.11: Performance of the STR position teleoperation for the second experiment. First two graphs: user position references in thin red line, actual position values of the STR tool on the workpiece surface (coordinates relative to the surface) in thick blue line, and position limits given by the 2D boundary constraint in dashed lines. Bottom graph: activation of the 2D boundary constraint for the position of the STR tool on the workpiece surface.

of the proposed AR-based teleoperation of the STR is sufficient for the task at hand.

Figure 6.12 shows several frames of the *third experiment*, which shows the performance of the 3D boundary and the WR reference hologram; see the video at (Video: Chapter 6, Experiment 3, 2022). Figure 6.12a shows how the user is commanding the WR and, when the WR end-effector approaches the 3D boundary, the boundary region closest to the WR end-effector is shown in blue and the warning sound is activated; see Figure 6.12b–d. Note that, when the user reference exceeds the 3D boundary, the WR end-effector is automatically kept within the allowed region. More details about this aspect can be further analyzed in Figure 6.13, which shows the allowed 3D region, the trajectory followed by the user reference, and the trajectory followed by

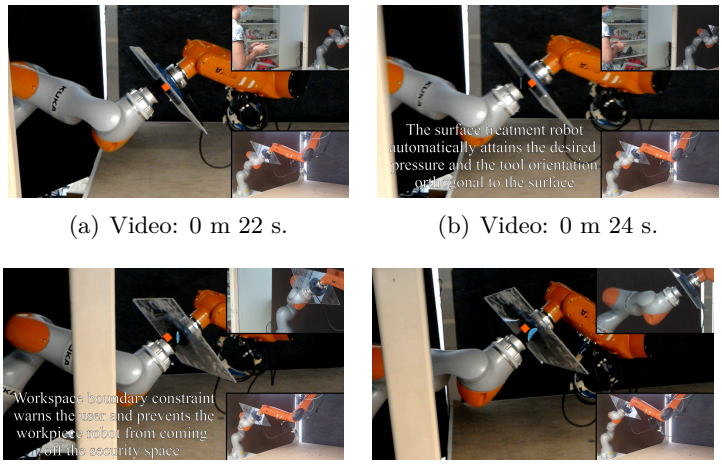
Table 6.2: Teleoperation errors for the 2D position $\overline{\mathbf{p}}_s$ of the STR tool on the workpiece surface.

	Position (cm)	
	x	y
Maximum deviation	1.8	3.2
Standard deviation	0.5	0.8

the WR end-effector.

Figures 6.14 and 6.15 show the position and orientation, respectively, followed by the workpiece, which are due to the WR teleoperation, together with the reference values provided by the user. In particular, it can be appreciated that the trajectory described by the workpiece corresponds closely to the user reference values, except obviously when the 3D boundary constraint is active; see the bottom graph in Figure 6.14. In fact, the maximum deviation of the actual workpiece position values compared to the user reference values, when the 3D boundary constraint was not active, was around 1.2 cm, with a standard deviation of around 0.4 cm; see Table 6.3. Moreover, the maximum deviation of the actual workpiece orientation values compared to the user reference values was around 1.7°, with a standard deviation of around 0.3°; see Table 6.3. Note that these teleoperation error values include all the potential sources of error: communication delays, high-level and low-level robot control, teleoperation system, etc. Therefore, it can be concluded that the accuracy of the proposed AR-based teleoperation of the WR is sufficient for the task at hand.

Figure 6.17 depicts several frames of the *fourth experiment*, which shows the performance of the overall system when the user commands simultaneously both robots using the proposed AR-based interface; see the video at (Video: Chapter 6, Experiment 4, 2022). Figure 6.17a–d show how the user modifies the orientation of the WR while, at the same time, commanding the STR tool towards one side of the workpiece. Note that, in this situation, when the WR end-effector is close to one side of the 3D boundary, it is partially shown by the corresponding blue hologram. Furthermore, Figure 6.17e shows how the user simultaneously commands both robots to reach both 2D and 3D boundaries,



(a) Video: 0 m 22 s. (b) Video: 0 m 24 s.

(c) Video: 1 m 04 s. (d) Video: 1 m 46 s

Figure 6.12: Third experiment: frames of the video showing the performance of the 3D boundary and the WR reference hologram. See the video at (Video: Chapter 6, Experiment 3, 2022).

Table 6.3: Teleoperation errors for the pose $\mathbf{p}_w$ (i.e., position and orientation) of the WR.

	Position (cm)			**Orientation (deg)**		
	x	y	z	α	β	γ
Maximum deviation	1.2	0.9	0.1	1.7	1.5	1.7
Standard deviation	0.4	0.3	0.03	0.3	0.2	0.2

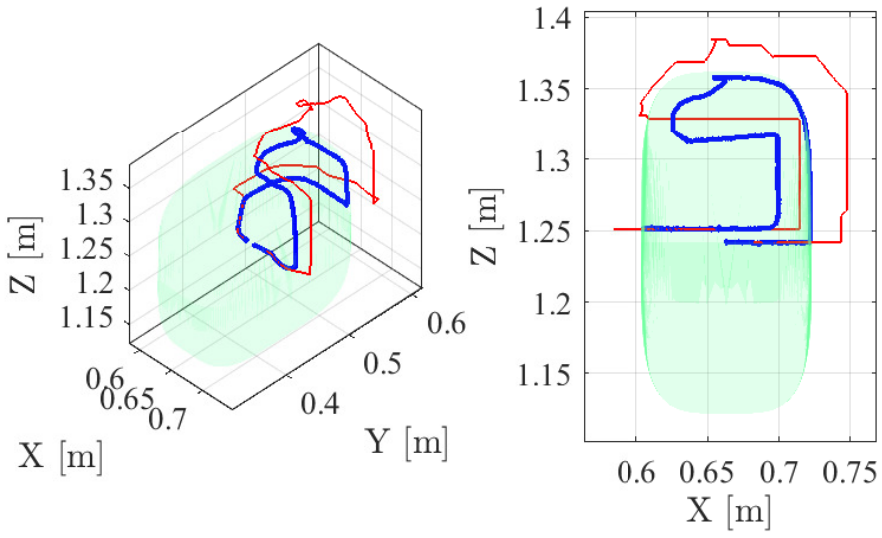


Figure 6.13: The 3D trajectory performance for the third experiment, showing the 3D boundary and the WR reference hologram (see the video at (Video: Chapter 6, Experiment 3, 2022)): 3D allowed region in green; trajectory followed by the user reference in thin red line; and trajectory followed by the WR end-effector in thick blue line.

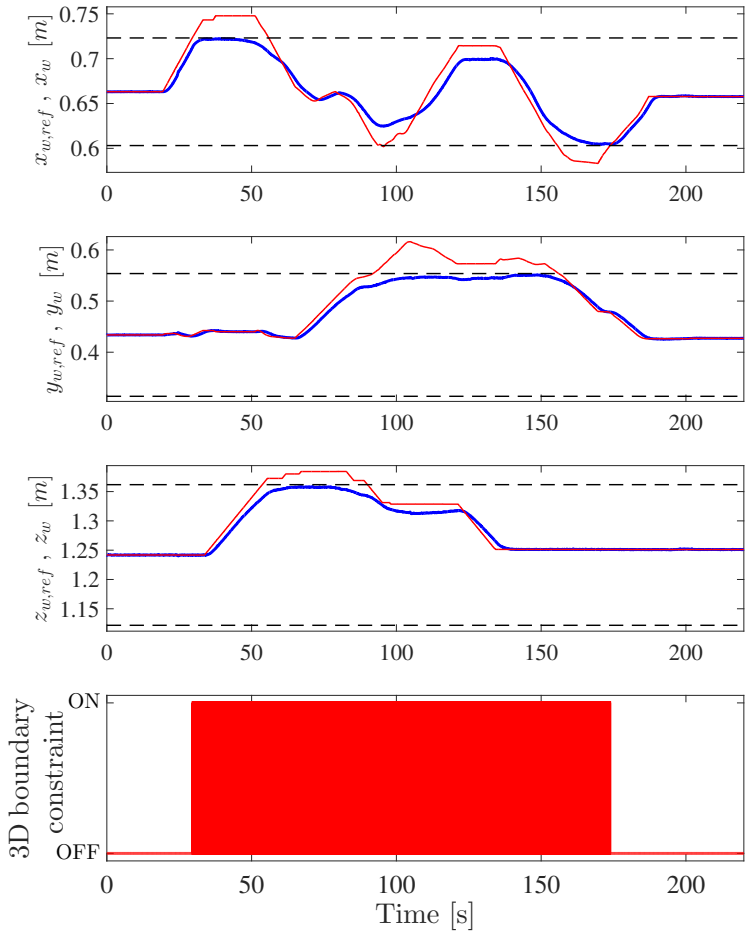


Figure 6.14: Performance of the WR position teleoperation for the third experiment. First three graphs: user position references in thin red line, actual position values of the workpiece in thick blue line, and position limits given by the 3D boundary constraint in dashed lines. Bottom graph: activation of the 3D boundary constraint for the workpiece position.

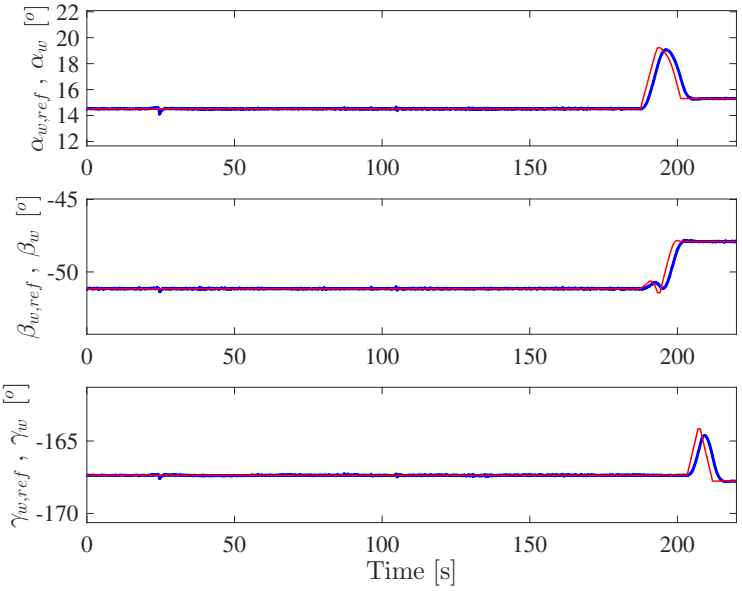
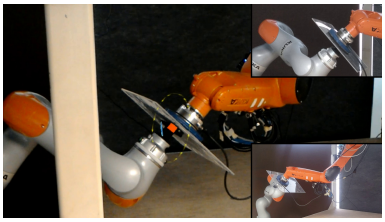


Figure 6.15: Performance of the WR angle teleoperation for the third experiment: user angular references in thin red line and actual angular values of the workpiece in thick blue line.

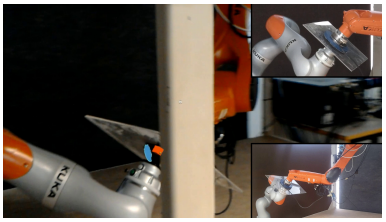
which are partially shown by the red and blue holograms, respectively. It is worth noting that, in addition to the mentioned holograms, the user hears different warning sounds. Figure 6.17f,g show how the user modifies again the orientation of the WR while, at the same time, commanding the STR tool towards the other side of the workpiece. Finally, Figure 6.17h shows how the STR tool reaches the 2D boundary while the user is also commanding the WR end-effector.

For the fourth experiment, Figure 6.18 shows the complete 2D trajectories followed by the user STR reference and the STR tool, whilst Figure 6.19 shows the complete 3D trajectories followed by the user WR reference and the WR end-effector. In both cases, as in the second and third experiments, the STR tool and the WR end-effector are automatically kept within the allowed regions despite the fact that, at some point, the user references exceed the 2D and 3D boundaries, respectively.

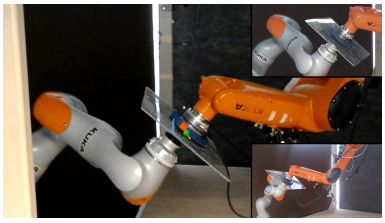
The teleoperation errors for the fourth experiment, in which the user commands simultaneously both robots using the proposed AR-based interface, are



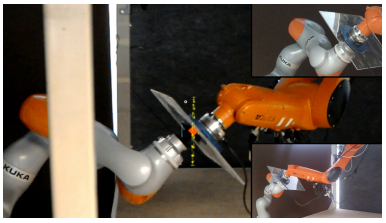
(a) Video: 1 m 19 s.



(b) Video: 1 m 44 s.

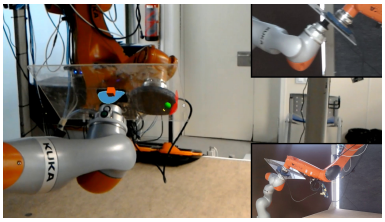


(c) Video: 1 m 54 s.

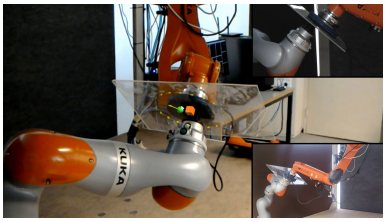


(d) Video: 1 m 55 s.

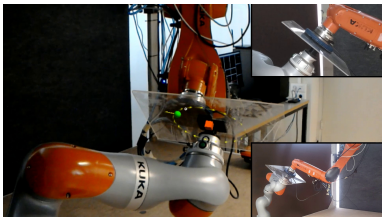
Figure 6.16: *Cont.*



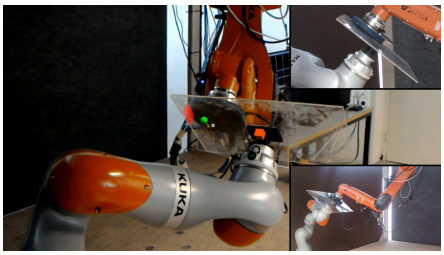
(a) Video: 2 m 13 s.



(b) Video: 2 m 42 s.



(c) Video: 2 m 52 s.



(d) Video: 3 m 05 s.

Figure 6.17: Fourth experiment: frames of the video showing the simultaneous teleoperation of both robots with the proposed AR-based interface. See the video at (Video: Chapter 6, Experiment 4, 2022).

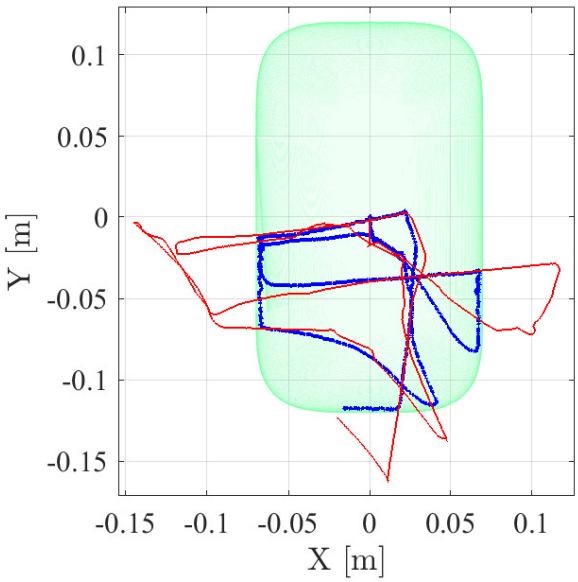


Figure 6.18: The 2D trajectory performance for the fourth experiment, showing the simultaneous teleoperation of both robots (see the video at (Video: Chapter 6, Experiment 4, 2022)): 2D allowed workpiece region in green; trajectory followed by the user reference in thin red line; and trajectory followed by the STR tool in thick blue line.

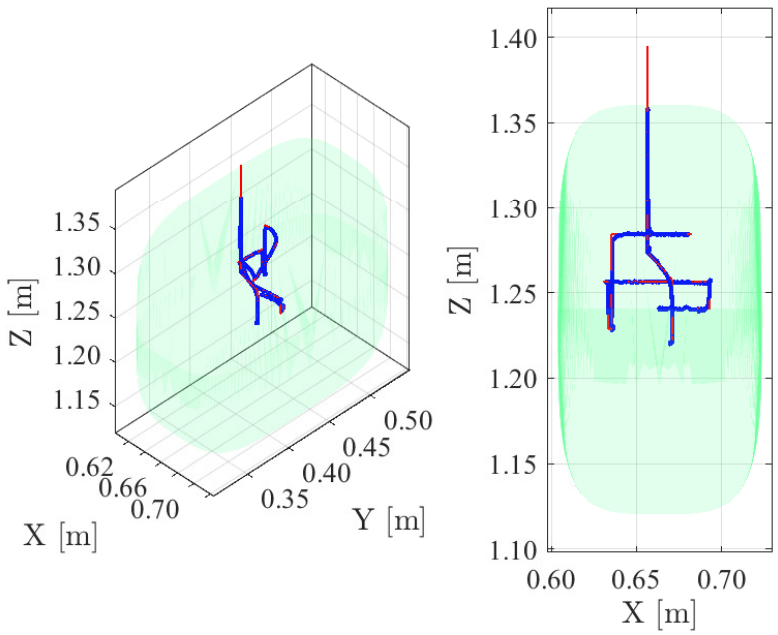


Figure 6.19: The 3D trajectory performance for the fourth experiment, showing the simultaneous teleoperation of both robots (see the video at (Video: Chapter 6, Experiment 4, 2022)): 3D allowed region in green; trajectory followed by the user reference in thin red line; and trajectory followed by the WR end-effector in thick blue line.

similar to those shown above for the second experiment (STR teleoperation) and third experiment (WR teleoperation): approximately 0.8 cm standard deviation for the position of the STR tool—see Table 6.2—and approximately 0.4 cm and 0.3° standard deviation for the WR position and orientation, respectively—see Table 6.3. As mentioned above, these teleoperation error values include all the potential sources of error: communication delays, high-level and low-level control of both robots, teleoperation system, etc. Therefore, it is concluded that the accuracy achieved by the proposed AR-based approach for teleoperating the bimanual robot system is satisfactory.

6.5 Conclusions

A solution to improve the assisted bimanual robot teleoperation has been developed in this work using augmented reality (AR) technology and tools. In particular, a new AR interface using the Microsoft® *HoloLens glasses* has been proposed to mitigate the problems in terms of user ergonomics and task performance (i.e., completion time and finishing quality) raised from the use of conventional PC-based user interfaces. In addition, this work has proposed and followed a new methodology to design and develop AR interfaces for bimanual robotic systems.

The effectiveness and applicability of the proposed AR interface were shown by means of real experimentation with an advanced bimanual robot application consisting of two robotic arms: a 7R cobot and a 6R industrial manipulator.

It is worth noting that several users, who tested both the conventional PC-based interface and the proposed AR interface, found the latter more intuitive and were able to conduct the robot teleoperation task faster. Note that when the users teleoperated the bimanual robot system using the conventional PC-based interface, most of them complained about the difficulty of checking whether the robots were performing the task correctly or not. In addition, the users indicated that with the conventional PC-based interface, it was not easy for them to command both robots simultaneously because they could not pay attention to so many reference signals shown. These facts negatively affected the performance of the users in terms of the time required to complete the task. Thus, the mentioned issues were mitigated with the proposed AR interface, significantly improving the user performance in the teleoperation task.

Another relevant remark is that the users also indicated that the warning sounds helped them in the early stages of the teleoperation task but, as the time of use of the interface increased, these sounds were annoying and they preferred only the visual warnings.

Chapter 7

Conclusions

7.1 Main Results

This work has proposed applications which address the problem of robust Human-Robot Interaction for surface treatment tasks, and these applications have been proved by experimentation results.

- In Chapter 3 the principles of Sliding Mode Control control and task priority based strategy of control used in this work are described and discussed.
- In Chapter 4 a complex Human-Robot Interaction, combining automatic and manual modes of operation, has been presented. This application ensures that the human operator can guide the polishing tool attached to the robot end-effector to manually treat arbitrary points of the workpiece surface, thanks to a hybrid admittance-SMC controller, while the robot system ensures the accomplishment of important constraints for the surface treatment task: orthogonality to the surface, a smooth approach towards the surface, limited pressure on it, all at the same time that it makes sure the tool does not abandon the allowed area, defined by a superellipsoid boundary. Moreover, the automatic mode is activated when the human user stops guiding the tool, so predefined points of the surface are treated by the robotic system, task which can be interrupted at any moment by the user in order to treat other areas of the surface. This is accomplished with the use of not only SMC and a task-priority based architecture, but also the integration of computer vision (in or-

der to locate the workpiece) and force-torque information, creating a truly cooperative application, which is significant, taking into account that previous works dealing with the automation of these industrial operations use a completely automatic operation of the robot system or consider a low degree of human-robot interaction. The effectiveness and feasibility of the proposed approach was shown with several experiments using a 6R robotic arm.

– In Chapter 5, conventional and non-conventional SMC and task-priority based architecture advantages for Human-Robot Interaction are generalized and applied to interaction between two robots and a human user. Thus, an application consisting in the assisted teleoperation of a bimanual robot system is presented, where one robot arm holds the workpiece while the other has the surface treatment tool attached to its end-effector. The bimanual system as a whole allows the human user to teleoperate some coordinates of both robots (position and workpiece orientation in one case, and 2D position of the tool on the workpiece surface in the other), while ensuring boundary constraints (so the workpiece does not abandon the allowed area or its orientation goes beyond the allowed limits; and so the tool does not go beyond the workpiece limits), constant pressure exerted by the tool over the workpiece, and perpendicularity of the tool to the workpiece surface, adapting itself to sudden rotations of the workpiece commanded by the operator. This application has been thoroughly validated through real experiments using an industrial 6R robot arm and a 7R cobot.

– In Chapter 6, the difficulties experimented when operating the application in 5 are used as a starting point in order to develop an Augmented Reality interface which improves the assisted teleoperation of bimanual systems, and a new methodology to develop augmented reality interfaces is presented, discussed and applied in the application itself. The application has been validated through real experimentation, including its test by several users who compared this AR interface with a conventional PC based interface for the application in 5, stating clearly that the AR interface was more intuitive and allowed them to perform the task faster.

7.2 Contributions

The research conducted for this PhD thesis has led to four publications in high impact international journals, which are summarized in this section.
From Chapter 4:

- A. García, V. Girbés-Juan, J. E. Solanes, L. Gracia, C. Perez-Vidal and J. Tornero, "Human-Robot Cooperation for Surface Repair Combining Automatic and Manual Modes," in IEEE Access, vol. 8, pp. 154024-154035, 2020,
  doi: 10.1109/ACCESS.2020.3014501.
  SCI-JCR (2020): Impact 3.367, Position 94/273 (Q2) in the category: ENGINEERING, ELECTRICAL & ELECTRONIC.

- Alberto García, Luis Gracia, J. Ernesto Solanes, Vicent Girbés-Juan, Carlos Perez-Vidal, Josep Tornero, Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints, Computers & Industrial Engineering, Volume 158, 2021, 107366, ISSN 0360-8352,
  https://doi.org/10.1016/j.cie.2021.107366.
  SCI-JCR (2021): Impact 7.180, Position 19/113 (Q1) in the category: COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS.

From Chapter 5:

- Alberto García, J. Ernesto Solanes, Luis Gracia, Pau Muñoz-Benavent, Vicent Girbés-Juan & Josep Tornero (2022) Bimanual robot control for surface treatment tasks, International Journal of Systems Science,
  DOI 10.1080/00207721.2021.1938279.
  SCI-JCR (2021): Impact 2.648, Position 39/109 (Q2) in the category: COMPUTER SCIENCE, THEORY & METHODS.

From Chapter 6:

- Alberto García, J.E. Solanes, Adolfo Muñoz, Luis Gracia and Josep Tornero. 2022. "Augmented Reality-Based Interface for Bimanual Robot Teleoperation". Applied Sciences 12, no.0: 4379.
  https://doi.org/10.3390/appl2094379.
  SCI-JCR (2021): Impact 2.838, Position 39/92 (Q2) in the category: ENGINEERING, MULTIDISCIPLINARY.

7.3 Further work

The human-machine interaction proposed and developed in this work poses a direct continuation: machine-learning. The operator's decisions when guiding or commanding the robots movements in the different applications presented in this work could be used as an input in order to increasingly automate the surface treatment process by finding patterns.

In a similar way, this information can also be used to improve the detection algorithms previous to the surface treatment; for instante, in the contributions presented in Chapter 4, the automated surface treatment sequence could be determined by a previous detection phase, and this detection phase could be refined with the information extracted from the manual intervention of the human user, in order to find out if there were defects missed by the detection algorithm.

And regarding teleoperation and Augmented-Reality, as presented in Chapters 5 and 6, this approach can be deepened with the development of applications in the path of the metaverse. The application of the mevateverse to the industrial environment offers a rich field of research, where multi-user virtual spaces could be set up to interact with the robotic systems while communicating with each other remotely.

In this way, complex shared control applications could be addressed remotely as a shared experience among several operators who could exchange ideas and transmit feedback to each other instantly while simultaneously operating the industrial facilities.

Bibliography

(Accessed 2020/06/10). Active contact flange (ACF) tool from FerRobotics. https://www.ferrobotics.com/en/technologie-produkte/produkte/acf/.

Abi-Farraj, F., Pacchierotti, C., Arenz, O., Neumann, G., and Giordano, P. R. (2020). A haptic shared-control architecture for guided multi-target robotic grasping. *IEEE Transactions on Haptics*, 13(2):270–285.

Astanin, S., Antonelli, D., Chiabert, P., and Alletto, C. (2017). Reflective workpiece detection and localization for flexible robotic cells. *Robotics and Computer-Integrated Manufacturing*, 44:190 – 198.

Baek, J., Jin, M., and Han, S. (2016). A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Transactions on Industrial Electronics*, 63(6):3628–3637.

Bandala, M., West, C., Monk, S., Montazeri, A., and Taylor, C. J. (2019). Vision-based assisted tele-operation of a dual-arm hydraulically actuated robot for pipe cutting and grasping in nuclear environments. *Robotics*, 8(2).

Bi, Z. and Wang, L. (2010). Advances in 3D data acquisition and processing for industrial applications. *Rob. Comput.-Integr. Manuf.*, 26(5):403 – 413.

Bian, F., Li, R., Zhao, L., Liu, Y., and Liang, P. (2018). Interface design of a human-robot interaction system for dual-manipulators teleoperation based on virtual reality. In *2018 IEEE International Conference on Information and Automation (ICIA)*, pages 1361–1366.

Brantner, G. and Khatib, O. (2021). Controlling ocean one: Human-robot collaboration for deep-sea manipulation. *Journal of Field Robotics*, 38(1):28–51.

Cao, L., Li, G., Xu, Y., Zhang, H., Shu, X., and Zhang, D. (2021). A brain-actuated robotic arm system using non-invasive hybrid brain-computer interface and shared control strategy. *JOURNAL OF NEURAL ENGINEERING*, 18(4).

Cardoso, J. C. S. (2016). Comparison of gesture, gamepad, and gaze-based locomotion for VR worlds. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 319–320, New York, NY, USA. Association for Computing Machinery.

Chatzilygeroudis, K., Fichera, B., Lauzana, I., Bu, F., Yao, K., Khadivar, F., and Billard, A. (2020). Benchmark for bimanual robotic manipulation of semi-deformable objects. *IEEE Robotics and Automation Letters*, 5(2):2443–2450.

Chen, H., Huang, P., and Liu, Z. (2019). Mode switching-based symmetric predictive control mechanism for networked teleoperation space robot system. *IEEE/ASME Transactions on Mechatronics*, 24(6):2706–2717.

Chen, H., Li, J., Wan, W., Huang, Z., and Harada, K. (2020a). Integrating combined task and motion planning with compliant control. *International Journal of Intelligent Robotics and Applications*, 4:149–163.

Chen, Y., Zhang, S., Wu, Z., Yang, B., Luo, Q., and Xu, K. (2020b). Review of surgical robotic systems for keyhole and endoscopic procedures: state of the art and perspectives. *Frontiers of Medicine*, 14:382–403.

Chiaverini, S., Oriolo, G., and Walker, I. (2008). Kinematically redundant manipulators. *Springer Handbook of Robotics*, pages 245–268.

Chu, Y. B. and Chang, C. W. (2021). A mobile augmented reality interface on additive manufacturing. In Zakaria, Z. and Emamian, S. S., editors, *Advances in Electrical and Electronic Engineering and Computer Science*, pages 1–12, Singapore. Springer Singapore.

Clark, J. P., Lentini, G., Barontini, F., Catalano, M. G., Bianchi, M., and O'Malley, M. K. (2019). On the role of wearable haptics for force feedback in teleimpedance control for dual-arm robotic teleoperation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5187–5193.

Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer-Verlag, Berlin, Germany, 2nd edition.

Craig, A. B. (2013). Chapter 2 - augmented reality concepts. In Craig, A. B., editor, *Understanding Augmented Reality*, pages 39 – 67. Morgan Kaufmann, Boston.

Cui, L., Wang, H., Liang, X., Wang, J., and Chen, W. (2020). Visual servoing of a flexible aerial refueling boom with an eye-in-hand camera. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, page In press (doi: 10.1109/TSMC.2019.2957992).

Da Silva, A., Dos Santos, D., Fernandes, A., Vilas Boas, J., and Garcia, L. (2020). High-level path planning for an autonomous sailboat robot using Q-Learning. *Sensors*, 20(6):1550.

Diao, S., Chen, X, ., and Luo, J. (2018). Development and experimental evaluation of a 3D vision system for grinding robot. *Sensors*, 18(9):1–20.

Dumitrescu, C., Costea, I.-M., and Semenescu, A. (2021). Using brain-computer interface to control a virtual drone using non-invasive motor imagery and machine learning. *APPLIED SCIENCES-BASEL*, 11(24).

Edwards, C. and Spurgeon, S. (1998). *Sliding Mode Control: Theory and Applications*. Taylor & Francis, UK, 1st edition.

Fei, Y., Shi, P., and Lim, C.-C. (2020). Neural network adaptive dynamic sliding mode formation control of multi-agent systems. *International Journal of Systems Science*, 51(11):2025–2040.

Ferraro, A., Indri, M., and Lazzero, I. (2012). Dynamic update of a virtual cell for programming and safe monitoring of an industrial robot. *IFAC Proceedings Volumes*, 45(22):822 – 827. 10th IFAC Symposium on Robot Control.

Gadre, S. Y., Rosen, E., Chien, G., Phillips, E., Tellex, S., and Konidaris, G. (2019). End-user robot programming using mixed reality. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2707–2713.

García, A., Girbés-Juan, V., Solanes, J. E., Gracia, L., Perez-Vidal, C., and Tornero, J. (2020). Human-robot cooperation for surface repair combining automatic and manual modes. *IEEE Access*, 8:154024–154035.

García, A., Gracia, L., Solanes, J. E., Girbés-Juan, V., Perez-Vidal, C., and Tornero, J. (2021). Robotic assistance for industrial sanding with a smooth approach to the surface and boundary constraints. *Computers & Industrial Engineering*, 158:107366.

García, A., Solanes, J. E., Gracia, L., Muñoz Benavent, P., Girbés-Juan, V., and Tornero, J. (2022a). Bimanual robot control for surface treatment tasks. *International Journal of Systems Science*, 53(1):74–107.

García, A., Solanes, J. E., Muñoz, A., Gracia, L., and Tornero, J. (2022b). Augmented reality-based interface for bimanual robot teleoperation. *Applied Sciences*, 12(9).

García, N., Rosell, J., and Suárez, R. (2019). Motion planning by demonstration with human-likeness evaluation for dual-arm robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2298–2307.

Garcia-Camacho, I., Lippi, M., Welle, M. C., Yin, H., Antonova, R., Varava, A., Borras, J., Torras, C., Marino, A., Alenyá, G., and Kragic, D. (2020). Benchmarking bimanual cloth manipulation. *IEEE Robotics and Automation Letters*, 5(2):1111–1118.

Garelli, F., Gracia, L., Sala, A., and Albertos, P. (2011). Sliding mode speed auto-regulation technique for robotic tracking. *Robotics and Autonomous Systems*, 59(7-8):519–529.

Garelli, F., Mantz, R., and De Battista, H. (2007). Sliding mode reference conditioning to preserve decoupling of stable systems. *Chemical Engineering Science*, 62:4705–4716.

Gaz, C., Magrini, E., and De Luca, A. (2018). A model-based residual approach for human-robot collaboration during manual polishing operations. *Mechatronics*, 55:234–247.

Girbés-Juan, V., Schettino, V., Demiris, Y., and Tornero, J. (2021). Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks. *IEEE Transactions on Haptics*, 14(1):44–56.

Golub, G. and Van Loan, C. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3rd edition.

Gorjup, G., Dwivedi, A., Elangovan, N., and Liarokapis, M. (2019). An intuitive, affordances oriented telemanipulation framework for a dual robot arm hand system: On the execution of bimanual tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3611–3616.

Gracia, L., Andres, J., and Tornero, J. (2009). Trajectory tracking with a 6R serial industrial robot with ordinary and non-ordinary singularities. *International Journal of Control, Automation and Systems*, 7(1):85–96.

Gracia, L., Sala, A., and Garelli, F. (2012). A path conditioning method with trap avoidance. *Robotics and Autonomous Systems*, 60(6):862–873.

Gracia, L., Solanes, J., Muñoz-Benavent, P., Valls Miro, J., Perez-Vidal, C., and Tornero, J. (2019). Human-robot collaboration for surface treatment tasks. *Interaction Studies*, 20:148–184.

He, Y., Liang, B., Yang, J., Li, S., and He, J. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors*, 17(8):1862–1–1862–16.

Hess, R. (2010). *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Focal Press. https://www.sciencedirect.com/book/9780240814308/blender-foundations.

Huang, S., Zhu, Z., Chen, J., Zhou, X., Yu, J., Gao, P., and Wang, H. (2020). Target force tracking and automatic contour surface processing in grinding of industrial robots. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 188–195.

Ibarguren, A., Eimontaite, I., Outón, J. L., and Fletcher, S. (2020). Dual arm co-manipulation architecture with enhanced human-robot communication for large part manipulation. *Sensors*, 20(21).

Ismail, A. W., Billinghurst, M., Sunar, M. S., and Yusof, C. S. (2019). Designing an augmented reality multimodal interface for 6dof manipulation techniques. In Arai, K., Kapoor, S., and Bhatia, R., editors, *Intelligent Systems and Applications*, pages 309–322, Cham. Springer International Publishing.

Isop, W. A., Gebhardt, C., Nägeli, T., Fraundorfer, F., Hilliges, O., and Schmalstieg, D. (2019). High-level teleoperation system for aerial exploration of indoor environments. *Frontiers in Robotics and AI*, 6:95.

Jackson, S. (2015). *Unity 3D UI Essentials*. Packt Publishing. https://dl.acm.org/citation.cfm?id=2789365.

Jin, Y., Chen, J., Zhang, S., Chen, W., and Zheng, X. (2019). *Invasive Brain Machine Interface System*, pages 67–89. Springer Singapore, Singapore.

Johnson, M. and Vera, A. (2019). No ai is an island: The case for teaming intelligence. *AI Magazine*, 40(1):16–28.

Joshi, R. P., Tarapure, J. P., and Shibata, T. (2020). Electric wheelchair-humanoid robot collaboration for clothing assistance of the elderly. In *2020 13th International Conference on Human System Interaction (HSI)*, pages 300–306.

Kapoor, A., Li, M., and Taylor, R. H. (2005). Spatial motion constraints for robot assisted suturing using virtual fixtures. In Duncan, J. S. and Gerig, G., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, pages 89–96, Berlin, Heidelberg. Springer Berlin Heidelberg.

Khan, A. H. and Li, S. (2020). Sliding mode control with PID sliding surface for active vibration damping of pneumatically actuated soft robots. *IEEE Access*, 8:88793–88800.

Khan, A. H., Li, S., Chen, D., and Liao, L. (2020). Tracking control of redundant mobile manipulator: An RRNN based metaheuristic approach. *Neurocomputing*, 400:272–284.

Kharidege, A., Ting, D., and Yajun, Z. (2017). A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot. *International Journal of Advanced Manufacturing Technology*, 93:3921 – 3934.

Kieselbach, K. K., Nöthen, M., and Heuer, H. (2019). Development of a visual inspection system and the corresponding algorithm for the detection and subsequent classification of paint defects on car bodies in the automotive industry. *Journal of Coatings Technology and Research*, 16:1033–1042.

Kitson, A., Hashemian, A. M., Stepanova, E. R., Kruijff, E., and Riecke, B. E. (2017). Comparing leaning-based motion cueing interfaces for virtual reality locomotion. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 73–82.

Kuo, Y., Huang, S., and Lan, C. (2019). Sensorless force control of automated grinding/deburring using an adjustable force regulation mechanism. In *2019 Int. Conference on Robotics and Automation (ICRA)*, pages 9489–9495.

Lee, J., Chang, P. H., and Jin, M. (2017). Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Transactions on Industrial Electronics*, 64(8):6796–6804.

Levant, A. (2003). Higher-order sliding modes, differentiation and output-feedback control. *Int. Journal of Control*, 76(9-10):924–941.

Levant, A. (2005). Quasi-continuous high-order sliding-mode controllers. *IEEE Transactions on Automatic Control*, 50(11):1812–1816.

Li, C., Fahmy, A., and Sienz, J. (2019). An augmented reality based human-robot interaction interface using kalman filter sensor fusion. *Sensors*, 19(20):4586.

Li, Y., Guo, S., and Mukai, T. (2019). Position adjustment control of a nursing-care robot holding a patient in its arms. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 976–981.

Li, Z., Huang, B., Ye, Z., Deng, M., and Yang, C. (2018). Physical human-robot interaction of a robotic exoskeleton by admittance control. *IEEE Transactions on Industrial Electronics*, 65(12):9614–9624.

Li, Z., Li, C., Li, S., and Cao, X. (2019). A fault-tolerant method for motion planning of industrial redundant manipulator. *IEEE Transactions on Industrial Informatics*, 16(12):7469–7478.

Li, Z. and Li, S. (2020). A sparse optimization based control method for manipulator with simultaneous potential energy minimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*.

Liang, J., Xu, Z., Zhou, X., Li, S., and Ye, G. (2020). Recurrent neural networks-based collision-free motion planning for dual manipulators under multiple constraints. *IEEE Access*, 8:54225–54236.

Lipton, J. I., Fay, A. J., and Rus, D. (2018). Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters*, 3(1):179–186.

Liu, Y., Su, W., Li, Z., Shi, G., Chu, X., Kang, Y., and Shang, W. (2019). Motor-imagery-based teleoperation of a dual-arm robot performing manipulation tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 11(3):414–424.

Lopez, E., Zollo, L., and Guglielmelli, E. (2013). Teleoperated control based on virtual fixtures for a redundant surgical system. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 450–455.

Lu, Z., Huang, P., and Liu, Z. (2018). Predictive approach for sensorless bimanual teleoperation under random time delays with adaptive fuzzy control. *IEEE Transactions on Industrial Electronics*, 65(3):2439–2448.

Lv, H., Yang, G., Zhou, H., Huang, X., Yang, H., and Pang, Z. (2020). Teleoperation of collaborative robot for remote dementia care in home environments. *IEEE Journal of Translational Engineering in Health and Medicine*, 8:1–10.

Makris, S., Tsarouchi, P., Matthaiakis, A.-S., Athanasatos, A., Chatzigeorgiou, X., Stefos, M., Giavridis, K., and Aivaliotis, S. (2017). Dual arm robot in cooperation with humans for flexible assembly. *CIRP Annals*, 66(1):13–16.

Martínez, S. and Ortega, J. e. a. (2013). An industrial vision system for surface quality inspection of transparent parts. *The International Journal of Advanced Manufacturing Technology*, 68(5-8):1123–1136.

McConachie, D., Dobson, A., Ruan, M., and Berenson, D. (2020). Manipulating deformable objects by interleaving prediction, planning, and control. *The International Journal of Robotics Research*, 39(8):957–982.

Microsoft Hololens (Accessed 03/10/2022). Hololens (1st gen) hardware details. https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details.

Microsoft Hololens (2nd gen) hardware details (Accessed 04/24/2020). https://www.microsoft.com/en-us/hololens/hardware.

Mistry, P. and Maes, P. (2009). Sixthsense: A wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, SIGGRAPH ASIA '09, New York, NY, USA. Association for Computing Machinery.

Mitash, C., Shome, R., Wen, B., Boularias, A., and Bekris, K. (2020). Task-driven perception and manipulation for constrained placement of unknown objects. *IEEE Robotics and Automation Letters*, 5(4):5605–5612.

Mohammad, A. E. K., Hong, J., and Wang, D. (2018). Design of a force-controlled end-effector with low-inertia effect for robotic polishing using macro-mini robot approach. *Robotics and Computer-Integrated Manuf.*, 49:54 – 65.

Mohsin, I., He, K., Cai, J., Chen, H., and Du, R. (2017). Robotic polishing with force controlled end effector and multi-step path planning. In *2017 IEEE Int. Conference on Information and Automation (ICIA)*, pages 344–348.

Mohsin, I., He, K., Li, Z., and Du, R. (2019). Path planning under force control in robotic polishing of the complex curved surfaces. *Applied Sciences*, 9(24):5489.

Muñoz, A., Mahiques, X., Solanes, J. E., Martí, A., Gracia, L., and Tornero, J. (2019). Mixed reality-based user interface for quality control inspection of car body surfaces. *Journal of Manufacturing Systems*, 53:75–92.

Muñoz, A., Martí, A., Mahiques, X., Gracia, L., Solanes, J. E., and Tornero, J. (2020). Camera 3D positioning mixed reality-based interface to improve worker safety, ergonomics and productivity. *CIRP Journal of Manufacturing Science and Technology*, 28:24 – 37.

Muñoz-Benavent, P., Solanes, J. E., Gracia, L., and Tornero, J. (2019). Robust auto tool change for industrial robots using visual servoing. *International Journal of Systems Science*, 50(2):432–449.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *Int. J. of Robotics Research*, 6(2):3–15.

Nemec, B., Yasuda, K., Mullennix, N., Likar, N., and Ude, A. (2018). Learning by demonstration and adaptation of finishing operations using virtual mechanism approach. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7219–7225.

Nenchev, D. (1995). Tracking manipulator trajectories with ordinary singularities: a null space-based approach. *The Int. Journal of Robotics Research*, 14(4):399–404.

Nicolis, D., Palumbo, M., Zanchettin, A. M., and Rocco, P. (2018). Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. *IEEE Robotics and Automation Letters*, 3(2):796–803.

Niemeyer, G., Preusche, C., Stramigioli, S., and Lee, D. (2016). *Telerobotics*, pages 1085–1108. Springer International Publishing, Cham.

Nilsen, M., Sikström, F., Christiansson, A.-K., and Ancona, A. (2019). Robust vision-based joint tracking for laser welding of curved closed-square-butt joints. *Int. J. of Advanced Manufacturing Technology*, 101(5):1967–1978.

Oba, Y. and Kakinuma, Y. (2017). Simultaneous tool posture and polishing force control of unknown curved surface using serial-parallel mechanism polishing machine. *Precision Engineering*, 49:24–32.

Ochoa, H. and Cortesão, R. (2019). Control architecture for robotic-assisted polishing tasks based on human skills. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 630–637.

Parigi Polverini, M., Zanchettin, A. M., and Rocco, P. (2019). A constraint-based programming approach for robotic assembly skills implementation. *Robotics and Computer-Integrated Manufacturing*, 59:69–81.

Park, S.-O., Lee, M. C., and Kim, J. (2020). Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics. *International Journal of Control, Automation and Systems*, 18(8):2095–2107.

Perez-Vidal, C., Gracia, L., Sanchez-Caballero, S., Solanes, J. E., Saccon, A., and Tornero, J. (2019). Design of a polishing tool for collaborative robotics using minimum viable product approach. *International Journal of Computer Integrated Manufacturing*, 32(9):848–857.

Qu, J., Zhang, F., Wang, Y., and Fu, Y. (2019). Human-like coordination motion learning for a redundant dual-arm robot. *Robotics and Computer-Integrated Manufacturing*, 57:379–390.

Rakita, D., Mutlu, B., Gleicher, M., and Hiatt, L. M. (2019). Shared control–based bimanual robot manipulation. *Science Robotics*, 4(30).

Rosen, E., Whitney, D., Phillips, E., Chien, G., Tompkin, J., Konidaris, G., and Tellex, S. (2019). Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *The International Journal of Robotics Research*, 38(12-13):1513–1526.

Salehian, S. S. M., Figueroa, N., and Billard, A. (2018). A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 37(10):1205–1232.

Santos, L. and Cortesão, R. (2018). Computed-torque control for robotic-assisted tele-echography based on perceived stiffness estimation. *IEEE Transactions on Automation Science and Engineering*, 15(3):1337–1354.

Saracino, A., Oude-Vrielink, T. J. C., Menciassi, A., Sinibaldi, E., and Mylonas, G. P. (2020). Haptic intracorporeal palpation using a cable-driven parallel robot: A user study. *IEEE Transactions on Biomedical Engineering*, 67(12):3452–3463.

Sato, A., Shen, K., Minami, M., and Matsuno, T. (2017). Improvement of force-sensorless grinding accuracy with resistance compensation. *Artificial Life and Robotics*, 22(1):509–514.

Segreto, T. and Teti, R. (2019). Machine learning for in-process end-point detection in robot-assisted polishing using multiple sensor monitoring. *Int. J. Adv. Manuf. Technol.*, 103(9):4173–4187.

Selvaggio, M., Abi-Farraj, F., Pacchierotti, C., Giordano, P. R., and Siciliano, B. (2018). Haptic-based shared-control methods for a dual-arm system. *IEEE Robotics and Automation Letters*, 3(4):4249–4256.

Selvaggio, M., Ghalamzan, A., Moccia, R., Ficuciello, F., and Siciliano, B. (2019). Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Sepúlveda, D., Fernández, R., Navas, E., Armada, M., and González-De-Santos, P. (2020). Robotic aubergine harvesting using dual-arm manipulation. *IEEE Access*, 8:121889–121904.

Siciliano, B. and Khatib, O. E. (2008). *Handbook of robotics*. Springer-Verlag, London, UK.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, UK.

Sintov, A., Macenski, S., Borum, A., and Bretl, T. (2020). Motion planning for dual-arm manipulation of elastic rods. *IEEE Robotics and Automation Letters*, 5(4):6065–6072.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., and Kragic, D. (2012). Dual arm manipulation-a survey. *Robotics and Autonomous Systems*, 60(10):1340–1353.

Solanes, J., Muñoz, A., Gracia, L., Martí, A., Girbés-Juan, V., and Tornero, J. (2020). Teleoperation of industrial robot manipulators based on augmented reality. *The International Journal of Advanced Manufacturing Technology*, 111:1077–1097.

Solanes, J. E., Gracia, L., Muñoz-Benavent, P., Valls Miro, J., Perez-Vidal, C., and Tornero, J. (2019). Robust Hybrid Position-Force Control for Robotic Surface Polishing. *ASME J. Manuf. Sci. Eng.*, 141(1):011013–1 – 011013–14.

Su, Y. and Zheng, C. (2020). A new nonsingular integral terminal sliding mode control for robot manipulators. *International Journal of Systems Science*, 51(8):1418–1428.

Suarez, A., Real, F., Vega, V. M., Heredia, G., Rodriguez-Castaño, A., and Ollero, A. (2020). Compliant bimanual aerial manipulation: Standard and long reach configurations. *IEEE Access*, 8:88844–88865.

Tang, G., Shi, Q., Zhang, Z., He, T., Sun, Z., and Lee, C. (2021). Hybridized wearable patch as a multi-parameter and multi-functional human-machine interface. *NANO ENERGY*, 81.

Taryudi and Wang, M.-S. (2018). Eye to hand calibration using anfis for stereo vision-based object manipulation system. *Microsystem Technologies*, 24(1):305–317.

Tedesco, A., Dallet, D., and Arpaia, P. (2021). Augmented reality (ar) and brain-computer interface (bci): Two enabling technologies for empowering the fruition of sensor data in the 4.0 era. In Di Francia, G. and Di Natale, C., editors, *Sensors and Microsystems*, pages 85–91, Cham. Springer International Publishing.

Tian, F., Li, Z., Lv, C., and Liu, G. (2016). Polishing pressure investigations of robot automatic polishing on curved surfaces. *The International Journal of Advanced Manufacturing Technology*, 87(1):639–646.

Tonin, L. and Millan, J. d. R. (2021). Noninvasive brain-machine interfaces for robotic devices. In Leonard, N., editor, *ANNUAL REVIEW OF CONTROL, ROBOTICS, AND AUTONOMOUS SYSTEMS, VOL 4, 2021*, volume 4 of *Annual Review of Control Robotics and Autonomous Systems*, pages 191–214.

Unity (Accessed 03/10/2022). Shaders core concepts. `https://docs.unity3d.com/Manual/Shaders.html`.

Utkin, V., Guldner, J., and Shi, J. (2009). *Sliding Mode Control in Electro-Mechanical Systems*. Taylor & Francis, London, 2nd edition.

Video: Chapter 4, Experiment 1 (Accessed 10/30/2020). `https://media.upv.es/player/?id=7449ae40-fc15-11ea-9ede-d1ad8f82e7cd`.

Video: Chapter 4, Experiment 2 (Accessed 10/30/2020). `https://media.upv.es/player/?id=ae62de20-fc11-11ea-9ede-d1ad8f82e7cd`.

Video: Chapter 4, Experiment 3 (Accessed 10/30/2020). `https://media.upv.es/player/?id=75ac5d70-fc13-11ea-9ede-d1ad8f82e7cd`.

Video: Chapter 5, Experiment 1 (2021). `https://media.upv.es/player/?id=036cf340-a731-11eb-a0b0-2fbcb59aaef7`.

Video: Chapter 5, Experiment 2 (2021). `https://media.upv.es/player/?id=cec9b0e0-a732-11eb-a0b0-2fbcb59aaef7`.

Video: Chapter 5, Experiment 3 (2021). `https://media.upv.es/player/?id=28a91da0-a731-11eb-a0b0-2fbcb59aaef7`.

Video: Chapter 5, Experiment 4 (2021). `https://media.upv.es/player/?id=42de19a0-a731-11eb-a0b0-2fbcb59aaef7`.

Video: Chapter 5, Experiment 5 (2021). https://media.upv.es/player/?id=e3904b60-a732-11eb-a0b0-2fbcb59aaef7.

Video: Chapter 5, Experiment 6 (2021). https://media.upv.es/player/?id=fa89cb70-a732-11eb-a0b0-2fbcb59aaef7.

Video: Chapter 5, Experiment 7 (2021). https://media.upv.es/player/?id=15ffabe0-a733-11eb-a0b0-2fbcb59aaef7.

Video: Chapter 6, Experiment 1 (2022). https://media.upv.es/player/?id=a64014f0-8a5a-11ec-ac0a-b3aa330d3dad.

Video: Chapter 6, Experiment 2 (2022). https://media.upv.es/player/?id=9504e6f0-8a61-11ec-b7c7-7d27dda7c5d5.

Video: Chapter 6, Experiment 3 (2022). https://media.upv.es/player/?id=17d88200-8f0b-11ec-be22-d786eca82090.

Video: Chapter 6, Experiment 4 (2022). https://media.upv.es/player/?id=29330720-8a8b-11ec-97cd-ab744f931636.

Villagrossi, E., Cenati, C., Pedrocchi, N., Beschi, M., and Molinari Tosatti, L. (2017). Flexible robot-based cast iron deburring cell for small batch production using single-point laser sensor. *The International Journal of Advanced Manufacturing Technology*, 92(1):1425–1438.

Wu, Q., Li, M., Qi, X., Hu, Y., Li, B., and Zhang, J. (2019). Coordinated control of a dual-arm robot for surgical instrument sorting tasks. *Robotics and Autonomous Systems*, 112:1–12.

Xu, B., Li, W., Liu, D., Zhang, K., Miao, M., Xu, G., and Song, A. (2022). Continuous hybrid bci control for robotic arm using noninvasive electroencephalogram, computer vision, and eye tracking. *MATHEMATICS*, 10(4).

Yao, B., Zhou, Z., Wang, L., Xu, W., Liu, Q., and Liu, A. (2018). Sensorless and adaptive admittance control of industrial robot in physical human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 51:158 – 168.

Yoon, H., Jeong, J. H., and Yi, B. (2018). Image-guided dual master-slave robotic system for maxillary sinus surgery. *IEEE Transactions on Robotics*, 34(4):1098–1111.

Yuan, J., Qian, Y., Gao, L., Yuan, Z., and Wan, W. (2019). Position-based impedance force controller with sensorless force estimation. *Assembly Automation*, 39(3):489–496.

Zhang, X., Ma, H., Luo, M., and Liu, X. (2020). Adaptive sliding mode control with information concentration estimator for a robot arm. *International Journal of Systems Science*, 51(2):217–228.

Zhao, J. and Allison, R. S. (2019). Comparing head gesture, hand gesture and gamepad interfaces for answering yes/no questions in virtual environments. *Virtual Reality*, pages 1–9.

Zhong, F., Wang, Y., Wang, Z., and Liu, Y. (2019). Dual-arm robotic needle insertion with active tissue deformation for autonomous suturing. *IEEE Robotics and Automation Letters*, 4(3):2669–2676.

Zimmermann, S., Hakimifard, G., Zamora, M., Poranne, R., and Coros, S. (2020). A multi-level optimization framework for simultaneous grasping and motion planning. *IEEE Robotics and Automation Letters*, 5(2):2966–2972.